

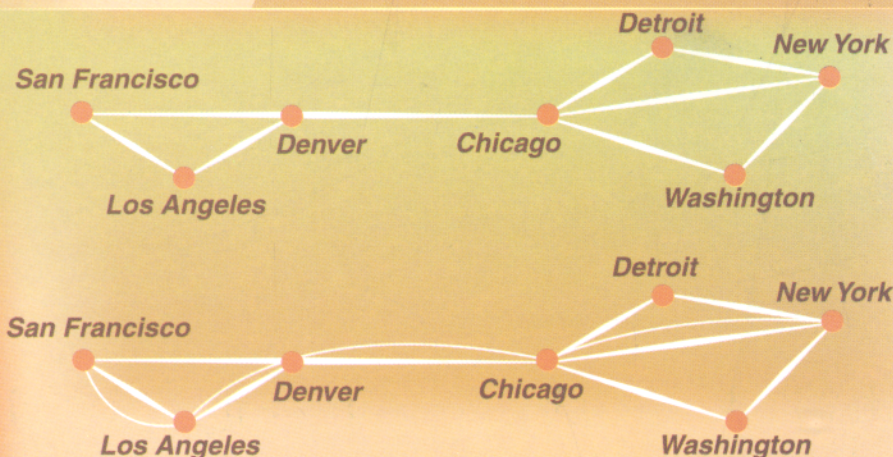
BÙI VIỆT HÀ

TỦ SÁCH



LẬP TRÌNH PASCAL

TẬP HAI



NHÀ XUẤT BẢN GIÁO DỤC

BÙI VIỆT HÀ

LẬP TRÌNH PASCAL

TẬP HAI

(Tái bản lần thứ ba)

NHÀ XUẤT BẢN GIÁO DỤC

LỜI NHÀ XUẤT BẢN

Trong những năm gần đây, việc đào tạo về Công nghệ Thông tin (CNTT) đã trở thành nhu cầu ngày càng cấp thiết hơn đối với mọi người, từ nhân viên bán hàng, kỹ sư, bác sĩ và phóng viên, đến nhà doanh nghiệp, cán bộ quản lý... Phù hợp với xu thế phát triển của xã hội, tin học đã trở thành một môn học được yêu thích ở nhà trường. Ngay từ trên ghế nhà trường không ít học sinh đã quyết tâm học tập để trở thành chuyên gia CNTT và một số em đã đạt thành tích cao tại các kỳ thi học sinh giỏi trong nước và quốc tế về tin học.

Để đáp ứng nhu cầu về sách tham khảo tin học cho học sinh phổ thông và góp phần nâng cao chất lượng đào tạo tin học trong nhà trường, Nhà xuất bản Giáo dục thành lập tủ sách "Tin học trong nhà trường" và sẽ lần lượt cho ra mắt bạn đọc loạt sách được chọn lọc có hệ thống cho học sinh phổ thông.

Chúng tôi rất mong nhận được ý kiến đóng góp của các thầy cô giáo, các em học sinh và đông đảo bạn đọc gần xa để ngày càng đáp ứng tốt hơn nhu cầu của bạn đọc.

Thư từ góp ý xin gửi về NXB Giáo dục, 81 Trần Hưng Đạo, Hà Nội.

Nhà xuất bản Giáo dục

LỜI NÓI ĐẦU

Đây là tập thứ hai của bộ sách gồm ba tập hướng dẫn lập trình trên ngôn ngữ Pascal cho học sinh phổ thông. Sách gồm hơn 250 bài tập và lời giải chi tiết và được chia thành 15 bài lớn. Các bài từ 1 đến 11 là phần cơ bản, các bài từ 12 đến 15 là phần nâng cao.

Phần cơ bản bao quát các vấn đề về kiểu mảng, kiểu tập hợp, kiểu bản ghi, kiểu file... Kiểu dữ liệu con trỏ, một trong các kiểu dữ liệu phức tạp nhất của Pascal nói riêng và các ngôn ngữ lập trình bậc cao nói chung cũng được đề cập ở đây. Trong phần này học sinh cũng bước đầu làm quen với khái niệm đồ thị, một trong những công cụ mạnh để giải quyết nhiều bài toán thực tế. Tiếp đó là các thuật toán và kỹ năng tạo hiệu ứng chuyển động trên màn hình đồ họa và một loạt các bài toán hình học hay, có thuật giải hữu hiệu được thể hiện thông qua các mô hình và các thuật toán đã được học. Bài cuối cùng trong phần cơ bản dùng để ôn luyện lại các dạng bài toán đã được học từ đầu quyển sách.

Phần nâng cao bao gồm 4 bài. Trong bài "Bài toán duyệt. Thuật toán quay lui" giới thiệu thuật toán quay lui, một thuật toán quen thuộc được sử dụng trong rất nhiều trường hợp để giải các bài toán khó. Đây là một trong các kỹ năng bắt buộc dành cho học sinh các lớp chuyên tin. "Bài toán tìm đường đi Euler" gồm các nội dung giới thiệu khái niệm, định nghĩa, tiêu chuẩn của đường đi và chu trình Euler trên đồ thị. Học sinh sẽ được làm quen với thuật toán tìm đường đi hay chu trình Euler và ứng dụng thuật toán này để giải quyết một loạt các bài toán tiêu biểu trên đồ thị. Tương tự như vậy với "Chu trình Hamilton" trong bài tiếp theo. Bài "Các bài toán trên đồ thị" ôn luyện các thuật toán và giải các bài toán tổng hợp kiến thức đã được học.

Hวัง vọng rằng bộ sách sẽ có ích cho các thầy cô giáo, các em học sinh và góp phần vào việc luyện thi học sinh giỏi Tin học.

Chúng tôi rất mong nhận được những ý kiến đóng góp để bộ sách được hoàn thiện hơn trong những lần xuất bản tới.

Tác giả

PHẦN I.

TÓM TẮT LÝ THUYẾT VÀ ĐỀ BÀI TẬP

Bài 1. Mảng nhiều chiều

A. Tóm tắt lý thuyết

Mảng là một kiểu dữ liệu có cấu trúc, bao gồm một số cố định các thành phần có cùng kiểu, có cùng một tên chung.

Khai báo mảng:

Tên_biến_mảng: ARRAY[chi_số_1,...,chi_số_n] OF kiểu_phần_tử;

+ Mảng hai chiều:

Ví dụ khai báo mảng A hai chiều có các phần tử thuộc kiểu nguyên như sau:

A: Array[1..5,1..5] of Integer;

Trong đó, số phần tử của mảng là $5 \times 5 = 25$, được sắp đặt trong bộ nhớ theo thứ tự sau:

A[1,1], A[1,2], ..., A[1,5];

A[2,1], A[2,2], ..., A[2,5];

.....;

A[5,1], A[5,2], ..., A[5,5];

Vì vậy mảng hai chiều còn được gọi là ma trận. Trong ví dụ, A là ma trận cấp 5×5 .

Để truy nhập tới phần tử ở hàng thứ i cột j ta dùng cách viết: A[i][j] hoặc A[i,j].

+ Các mảng nhiều chiều hơn được suy diễn một cách tự nhiên từ các mảng một chiều và hai chiều. Ví dụ, để khai báo một mảng ba chiều B có các phần tử thuộc tập các số nguyên, ta viết như sau:

B: Array [1..2,1..3,1..2] of Integer;

Số phần tử của mảng B là $2 \times 3 \times 2 = 12$, được sắp xếp trong bộ nhớ theo thứ tự sau:

B[1,1,1]; B[1,1,2];

B[1,2,1]; B[1,2,2];

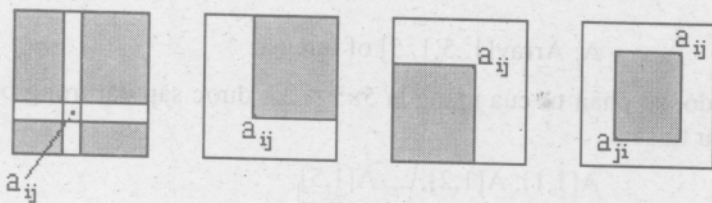
$B[1,3,1]; B[1,3,2];$
 $B[2,1,1]; B[2,1,2];$
 $B[2,2,1]; B[2,2,2];$
 $B[2,3,1]; B[2,3,2];$

Các bài tập sau chủ yếu ứng dụng trên mảng hai chiều.

B. Bài tập

Trong các bài tập từ 1 đến 11, kích thước của mảng và mảng được nhập từ tệp văn bản INPUT.DATA có dạng sau: dòng đầu tiên ghi số n là kích thước ma trận, n dòng tiếp theo ghi lần lượt các số hạng của hàng tương ứng, mỗi số cách nhau bởi dấu cách. Kết quả của chương trình đưa ra màn hình.

1. Cho mảng vuông các số thực $A = (a_{ij})$ kích thước $n \times n$. Hãy lập mảng $B = (b_{ij})$ cùng kích thước, biết b_{ij} bằng tổng các phần tử của A nằm trong phần được tô đậm.

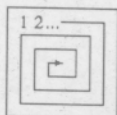


2. Cho mảng vuông các số thực $A = (a_{ij})$. Hãy tìm cách đổi chỗ các hàng và đổi chỗ các cột của ma trận sao cho phần tử lớn nhất của mảng nằm ở góc trái trên.

3. Cho mảng vuông các số thực $A = (a_{ij})$. Hãy tìm cách đổi chỗ các hàng và đổi chỗ các cột của ma trận sao cho phần tử lớn nhất của mảng nằm ở góc trái trên, phần tử nhỏ nhất của mảng nằm ở góc phải dưới.

4. Cho mảng vuông các số thực $A = (a_{ij})$. Hãy lập mảng mới từ A bằng cách bỏ đi hàng và cột có chứa phần tử có trị tuyệt đối lớn nhất.

5. Hãy lập mảng vuông kích thước 7×7 từ các số $1, 2, 3, \dots, 49$ biết các số sắp xếp trong mảng theo thứ tự sau:



6. Cho mảng số thực vuông A kích thước 7×7 . Hãy lập mảng một chiều độ dài 49 gồm các phần tử của mảng A theo thứ tự đã chỉ ra ở bài trên.

7. Cho mảng số thực vuông A kích thước 8×8 . Hãy lập mảng một chiều độ dài 64 gồm các phần tử của mảng A theo thứ tự sau:

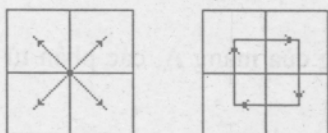


8. Cho mảng số thực vuông A kích thước $n \times n$. Hãy lập mảng một chiều độ dài n^2 gồm các phần tử của mảng A theo thứ tự giống như bài 7.

9. Cho mảng số thực vuông A kích thước $n \times n$. Hãy lập mảng một chiều độ dài n^2 gồm các phần tử của mảng A theo thứ tự giống bài 5.

10. Hãy lập mảng vuông kích thước $n \times n$ từ các số $1, 2, 3, \dots, n^2$, biết các số sắp xếp trong mảng theo thứ tự như bài 5.

11. Cho mảng số thực vuông A kích thước $2n \times 2n$. Hãy lập các mảng mới bằng cách đổi chỗ các khối vuông kích thước $n \times n$ của A theo các cách sau:



12. Hãy lập chương trình xây dựng các mảng sau (n nhập từ bàn phím):

$$\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ & & \dots & \\ 0 & 0 & \dots & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & \dots & 0 & 1 \\ 0 & \dots & 2 & 0 \\ & \dots & & \\ n & \dots & 0 & 0 \end{bmatrix}$$

13. Hãy lập chương trình xây dựng các mảng sau (n nhập từ bàn phím):

$$\begin{bmatrix} 1 & 2 & \dots & n-1 & n \\ 2 & 3 & \dots & n & n-1 \\ 3 & 4 & \dots & n-1 & n-2 \\ \dots & \dots & \dots & \dots & \dots \\ n & n-1 & \dots & 2 & 1 \end{bmatrix} \quad \begin{bmatrix} \frac{1}{1!} & \frac{1}{2!} & \dots & \frac{1}{n!} \\ \frac{1}{1!2} & \frac{1}{2!2} & \dots & \frac{1}{n!2} \\ \dots & \dots & \dots & \dots \\ \frac{1}{1!n} & \frac{1}{2!n} & \dots & \frac{1}{n!n} \end{bmatrix}$$

14. Cho bảng kết quả bóng đá là một mảng vuông $A = (a_{ij})$ kích thước $n \times n$, trong đó n là số đội bóng, $a_{ij} = 0$ khi đội i thua đội j , $a_{ij} = 1$ khi đội i hoà đội j và $a_{ij} = 3$ khi đội i thắng đội j .

a. Hãy tính điểm của các đội bóng.

Kết quả thể hiện ra màn hình theo dạng sau:

1: 30 điểm

2: 25

3: 34

.....

b. Hãy lập lại bảng kết quả mới, trong đó các đội được sắp xếp theo thứ tự đội kết quả cao đứng trước, đội kết quả thấp đứng sau.

Bảng kết quả là các text file có qui cách như sau:

+ Dòng đầu tiên là số n .

+ n dòng tiếp theo là n hàng của mảng A , các phần tử phân cách nhau bởi dấu cách.

15. Cho số tự nhiên n và mảng số thực A kích thước $4 \times n$ chứa dữ liệu về n hình chữ nhật: $(a[1,i], a[2,i])$ và $(a[3,i], a[4,i])$ là các tọa độ góc trái trên và góc phải dưới của hình chữ nhật thứ i . Hỏi có điểm nào trong mặt phẳng tọa độ thuộc tất cả các hình chữ nhật đã cho không? Hãy lập chương trình để giải bài toán đó.

Đầu vào là text file có qui cách như sau:

+ Dòng đầu tiên là số n .

+ n dòng tiếp theo, mỗi dòng gồm 4 số là các tọa độ $a[1,i]$, $a[2,i]$, $a[3,i]$, $a[4,i]$ tương ứng, các số phân cách nhau bởi dấu cách.

Kết quả thể hiện trên màn hình.

Bài 2. Các thao tác trên chuỗi ký tự

A. Tóm tắt lý thuyết

1. Khai báo kiểu chuỗi ký tự

Tên biến: `STRING[N]`;
Trong đó: `N` là số ký tự cực đại có thể của chuỗi.
Chiều dài tối đa của một chuỗi là 255.

Nếu trong khai báo không ghi `[N]` thì chuỗi có độ dài ngầm định là 255.

2. Các thủ tục và hàm trên chuỗi ký tự

- + **Delete**(`S, m, n`); thủ tục này sẽ xóa khỏi chuỗi `S` `n` ký tự bắt đầu từ vị trí thứ `m` (tính từ trái sang phải).
- + **Insert**(`S2, S1, m`); thủ tục chèn chuỗi `S2` vào `S1` ở vị trí thứ `m`.
- + **STR**(`n, S`); đổi giá trị số `n` thành chuỗi rồi gán cho `S`.
- + **VAL**(`S, n, Code`); đổi chuỗi chữ số `S` (`S` biểu diễn một số nguyên hay số thực nào đó) thành một số và gán cho biến `n`. `Code` là số nguyên để phát hiện lỗi: nếu phép biến đổi đúng thì `Code` có giá trị là 0, nếu sai là do `S` không biểu diễn đúng số nguyên hay số thực. `Code` sẽ nhận giá trị bằng vị trí của ký tự sai trong chuỗi.
- + **LENGTH**(`S`); hàm cho độ dài của biểu thức chuỗi ký tự `S`.
- + **COPY**(`S, m, n`); hàm cho một chuỗi mới bằng cách chép `n` ký tự từ chuỗi `S` bắt đầu từ vị trí `m`.
- + **CONCAT**(`S1, S2, ..., Sn`); hàm này sẽ ghép nối tất cả các chuỗi ký tự `S1, S2, ..., Sn` thành một chuỗi theo thứ tự đã viết.
- + **POS**(`S1, S2`); hàm cho vị trí đầu tiên xuất hiện chuỗi `S1` trong chuỗi `S2`. Nếu không tìm thấy hàm có giá trị là 0.

B. Bài tập

1. Cho 3 mảng một chiều `Hodem`, `Ten` và `Lop` cùng độ dài `n` chứa một danh sách `n` học sinh, dữ liệu về mỗi học sinh gồm họ đệm, tên và lớp là 3 phần tử cùng chỉ số trong 3 mảng. Hãy tạo mảng `DanhSach` cùng kích thước chứa danh sách đó, mỗi học sinh chiếm một phần tử của mảng. Dữ liệu của các mảng `Hodem`, `Ten` và `Lop` được cho trong các tệp văn

bản có tên tương ứng HODEM.TXT, TEN.TXT và LOP.TXT, mỗi dòng một phần tử của mảng. Kết quả của mảng Danhsach đưa ra màn hình.

2. Nhập một danh sách học sinh, mỗi học sinh là một xâu kí tự gồm họ tên và lớp theo mẫu sau: Nguyen Van Minh 12C. Hãy chứa danh sách vào 3 mảng: Holot, Ten và Lop, mỗi học sinh chiếm 3 phần tử cùng chỉ số trong 3 mảng. Dữ liệu đưa vào từ tệp văn bản DANHSACH, kết quả đưa ra màn hình.

3. Cho một đoạn văn chỉ gồm các chữ cái, các dấu cách và các dấu ngắt câu. Hãy sửa lại đoạn văn theo yêu cầu sau:

- Không được có 2 dấu cách đứng liền nhau.
- Dấu ngắt câu phải đứng sau chữ cái và trước dấu cách.
- Sau các dấu chấm câu phải viết hoa.

Dữ liệu vào cho bởi tệp ALPHABET.TXT, kết quả đưa ra tệp KQ.TXT.

4. Nhập vào từ bàn phím một đoạn văn bản là một xâu kí tự gồm các từ phân cách nhau bởi các dấu cách. Hãy sắp xếp lại các từ trong đoạn văn này theo thứ tự của bảng chữ cái tiếng Anh. Ví dụ đoạn văn bản ban đầu là "Mean City Town Food" thì kết quả sẽ là xâu "City Food Mean Town". Kết quả thể hiện trên màn hình.

5. Khi soạn thảo văn bản ta thường phải giống hàng. Để giống hàng phải, người ta thường tăng các dấu cách trong hàng.

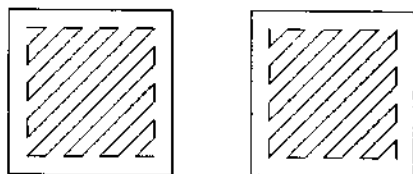
Cho text file f chứa một đoạn văn bản, mỗi dòng có ít nhất 2 từ. Hãy giống hàng cho văn bản đó. Kết quả ghi vào file g. Yêu cầu các nhóm dấu cách trong mỗi dòng không được khác biệt nhau quá 1 dấu cách (ngoại trừ nhóm dấu cách đầu dòng).

6. Nhập vào từ bàn phím một xâu kí tự là một số La Mã. Kiểm tra xem số liệu được nhập vào có đúng hay không. Kết quả thể hiện trên màn hình.

7. Để mã hoá một văn bản gồm toàn chữ cái tiếng Anh người ta có thể làm như sau: Thay mỗi kí tự bằng kí tự đứng sau nó trong bảng chữ cái, riêng z thay bằng a. Ví dụ 'anh' được mã hoá thành 'boi'. Bạn hãy lập chương trình để:

- a. Mã hoá một văn bản.
- b. Giải mã một văn bản đã mã hoá.

8. Để mã hoá một văn bản gồm n^2 kí tự, ta có thể chép văn bản đó vào một bảng vuông kích thước $n \times n$ theo một thứ tự nào đó, sau đó đọc theo một thứ tự khác. Chẳng hạn ta có thể chép và đọc văn bản theo các thứ tự ở hình sau:



Ví dụ nếu $n=3$ và văn bản là 'Em hoc' thì văn bản đã mã hoá là 'E meoh'.
Bạn hãy mã hoá các đoạn văn bản sau:

'I am a girl'.

'I am a student'.

Nếu độ dài của đoạn văn bản không phải là số chính phương, bạn hãy thêm một số dấu cách vào cuối văn bản.

9. Để mã hoá một văn bản gồm toàn chữ cái tiếng Anh người ta có thể làm như sau: Xếp bảng chữ cái theo vòng tròn, sau đó thay mỗi kí tự bằng kí tự đứng sau nó n vị trí trong bảng vòng tròn. Ví dụ với $n = 7$ thì 'anh' được mã hoá thành 'huo'. Bạn hãy lập chương trình nhập số tự nhiên n , sau đó:

1. Mã hoá một văn bản (xâu kí tự).
2. Giải mã một văn bản đã mã hoá.

10. Với cách mã hoá đã chỉ ra ở bài trên, bạn hãy lập chương trình:

- a. Mã hoá một văn bản (xâu kí tự).
- b. Giải mã một văn bản đã mã hoá.

11. Bạn hãy đề xuất một cách mã hoá khác dựa vào ý tưởng "ghi-đọc" như bài trên. Sau đó hãy lập chương trình:

- a. Mã hoá một văn bản (xâu kí tự).
- b. Giải mã một văn bản đã mã hoá.

12. Cho trước khoá là một hoán vị của n số $12 \dots n$. Khi đó để mã hoá một xâu kí tự ta có thể chia xâu thành từng nhóm n kí tự (riêng nhóm cuối cùng nếu không đủ n kí tự thì ta có thể thêm các dấu cách vào cho đủ) rồi hoán vị các kí tự trong từng nhóm. Chẳng hạn với khoá 3241 thì ta có thể mã hoá xâu 'english' thành 'gnlehs i'.

Với khoá như trên, bạn hãy mã hoá các đoạn văn bản sau:

'I am a girl'.

'I am a student'.

13. Hãy lập chương trình nhập một khoá (hoán vị của n số). Sau đó kiểm tra khoá và:

- a. Mã hoá một văn bản nhập từ bàn phím.
- b. Giải mã một văn bản đã mã hoá nhập từ bàn phím.

14. Để nâng cao độ tin cậy khi truyền các bit thông tin, mỗi bit được lặp lại 3 lần, ví dụ '011' được truyền thành '000111111'. Khi nhận thông tin, cứ một đoạn 3 bit được giải mã thành 1 bit là số gấp trong đoạn đó ít nhất 2 lần.

Ví dụ nếu thông tin nhận được là '000110010011' thì thông tin đã giải mã là '0101'. Cho xâu nhị phân là thông tin nhận được. Hãy viết chương trình giải mã thông tin đó.

15. Bạn hãy viết chương trình nhập vào một xâu kí tự, trong đó, nếu tính từ phải sang trái thì kí tự # là dấu hiệu xoá đi một kí tự đứng ngay trước nó nếu có. Ví dụ: '#Ta#ôi đi ngủ#h###học' có nghĩa là 'Tôi đi học'. Sau đó, in ra xâu đã sửa lại theo qui định trên.

Bài 3. Kiểu tập hợp

A. Tóm tắt lý thuyết

1. Định nghĩa

Dữ liệu kiểu tập hợp là một tập hợp những dữ liệu cùng thuộc một kiểu vô hướng đếm được (kiểu cơ sở của kiểu tập hợp).

Dữ liệu thuộc kiểu tập hợp được khai báo dưới dạng sau:

SET OF kiểu_cơ_sở ;

Ví dụ:

Chuso : SET OF 1..9;

mau : SET OF (Đỏ, vàng, xanh, tím);

2. Các phép toán trên tập hợp

+ **Các phép toán quan hệ:** Cho A và B là hai dữ liệu thuộc kiểu tập hợp, ta có các phép toán sau:

$A = B$: cho giá trị là True nếu hai tập hợp A, B là bằng nhau.

$A \neq B$: cho giá trị là True nếu hai tập hợp A, B là khác nhau.

$A \subseteq B$: cho giá trị là True nếu A là tập con của B.

$A \supseteq B$: cho giá trị là True nếu B là tập con của A.

Chú ý: Không có các phép toán $<$, $>$ trên tập hợp.

+ **Phép toán IN:** dùng để kiểm tra xem một phần tử nào đó có nằm trong một tập hợp không. Nếu phần tử đó nằm trong tập hợp thì phép toán trả về giá trị True, ngược lại cho giá trị False.

+ **Phép toán hợp, giao, hiệu:**

$A + B$: là hợp của A và B, cho tập hợp các phần tử thuộc A hoặc thuộc B.

$A * B$: là giao của A và B, cho tập hợp các phần tử thuộc A và thuộc B.

$A - B$: là hiệu của A và B, cho tập hợp các phần tử thuộc A nhưng không thuộc B.

Ví dụ: $A = [1, 2, 3]$; $B = [4, 5, 6]$. Khi đó ta có:

$A + B = [1, 2, 3, 4, 5, 6]$;

$A * B = []$; giao của A và B là tập rỗng.

$A - B = [1, 2, 3]$;

B. Câu hỏi và bài tập

1. Các khai báo sau đây đúng hay sai, nếu sai hãy chỉ rõ vì sao:

- a: set of byte;
- b: set of integer;
- c: set of string[10];
- d: set of record;
- e: set of set;
- f: set of char;
- g: set of 100..300;
- h: set of shortint;

2. Các toán tử sau đây có tương đương hay không:

- a. $P \text{ in } [0, 5, 19]$ và $(P=0) \text{ or } (P=5) \text{ or } (P=19)$.
- b. $P \text{ in } [20.. 50]$ và $(P \geq 20) \text{ and } (P \leq 50)$.

3. Trong các biểu thức sau, biểu thức nào chỉ đúng một tập hợp:

- a. $[3, 7, -1, 10]$
- b. $[267, 1567, 5.7, 67899]$
- c. $[0..0]$
- d. $[67, 1.2, \text{sqrt}(9)]$
- e. $[1..67, 100..150]$
- f. $[[3,5],[],[9]]$
- g. $[i = 4, j = 5, z = 6]$
- h. Set of Char
- i. Set of Byte

4. Cho chương trình sau:

```
Program P20304;  
Var thoat: set of char = ['e', 'E'];  
BEGIN  
  Write('Hay go E de thoat khoi chuong trinh: ');  
  Repeat ch:=readkey; Until ch in thoat;  
END.
```

Hãy tìm và sửa lỗi trong chương trình đó.

5. Bạn hãy viết hàm $\text{card}(A)$ đếm số phần tử của tập hợp A cho trước có kiểu set of 0..99.

6. Chương trình sau đây mô tả việc duyệt các phần tử của một tập hợp và liệt kê các phần tử đó ra màn hình:

```
Program P20306;  
  Var  tap: set of char = ['a'..'z'];  
       ch: char;  
  
  BEGIN  
  
    For ch:='a' to 'z' do  
  
      If ch in tap then Write(ch, ' ');  
  
    Readln;  
  
  END.
```

Hãy chạy thử chương trình.

7. Bạn hãy lập chương trình liệt kê các phần tử của một tập hợp số ra màn hình tương tự chương trình trên. Tập hợp số ban đầu được ghi trong tệp NUMBERS.DAT, mỗi số trên một dòng.

8. Chương trình sau minh họa việc gán giá trị cho một biến tập và một vài phép toán trên tập hợp như thêm phần tử vào tập hợp (phép hợp), loại bớt phần tử khỏi tập hợp (phép trừ):

```
Program P20308;  
  Var  tap: set of char;  
       ch: char;  
  
  BEGIN  
    tap:=['a','b','c'];  
    tap:=tap+['d'];  
    tap:=tap-['a'];  
    For ch:='a' to 'z' do  
      If ch in tap then Write(ch, ' ');  
    Readln;  
  
  END.
```

Bạn hãy chạy thử chương trình.

9. Bạn hãy lập chương trình tạo một tập hợp các số nguyên chẵn kiểu byte và loại khỏi nó các số chia hết cho 3. Kết quả thể hiện trên màn hình.

10. Chương trình sau minh hoạ việc dùng một tập hợp để kiểm tra việc nhập các giá trị cho một biến từ bàn phím:

```
Program P20311;  
Uses Crt;  
Var thoat: set of char;  
    ch: char;  
BEGIN  
    thoat:= ['e', 'E'];  
    Write('Hay go phim E de thoat khoi chuong trinh: ');  
    Repeat ch:=readkey; Until ch in thoat;  
END.
```

Hãy chạy thử chương trình.

11. Bạn đã biết dùng "sàng Erathosten" để tìm các số nguyên tố. Tuy nhiên khi đó chúng ta loại bỏ (sàng) các hợp số bằng cách đánh dấu. Bây giờ bạn hãy lập một chương trình khác dùng "sàng Erathosten" tìm các số nguyên tố kiểu byte. Yêu cầu trong chương trình phải dùng một tập hợp làm "sàng" để "đựng" các số nguyên và "sàng" khỏi nó các hợp số. Kết quả thể hiện trên màn hình.

12. Bạn hãy lập chương trình hiển thị một menu dạng sau trên màn hình:

1. Xem
2. Sua chua
3. Loai bo
4. Nhap them
5. Thoat

Lua chon cua ban: _

Sau đó đợi gõ một phím. Chương trình phải đợi cho tới khi phím gõ vào là một trong các chữ số 1..5 hoặc các chữ cái đầu của các tùy chọn thì thông báo phím gõ vào hợp lệ và kết thúc chương trình. Trong chương trình phải dùng một tập hợp để kiểm tra việc nhập giá trị cho biến từ bàn phím bằng cách tương tự như chương trình trên.

13. Hãy lập chương trình nhập vào một xâu kí tự từ bàn phím. Yêu cầu các kí tự nhập vào phải là các chữ cái thuộc bảng chữ cái tiếng Anh, bỏ qua các phím khác.

14. Hãy lập chương trình nhập vào một xâu nhị phân. Các kí tự nhập vào không hợp lệ bị bỏ qua.

15. Hãy lập chương trình nhập vào một số nguyên dương n và một dãy số là hoán vị của các số $1, 2, \dots, n$. Chương trình sẽ phải kiểm tra ngay xem từng số nhập vào có hợp lệ hay không. Cụ thể:

- Số n phải nằm trong khoảng từ 1 đến 125.
- Một số mới nhập vào của dãy phải nằm trong khoảng từ 1 đến n và không được trùng với số đã có trong dãy.

Nếu dữ liệu nhập vào không hợp lệ thì phải nhập lại cho tới khi đạt yêu cầu.

16. Chương trình sau minh họa cách thêm phần tử vào tập hợp trực tiếp từ bàn phím:

Program P20316;

Uses Crt;

Var tap: set of char;

ch: char;

BEGIN

tap:=[];

Write('Nhập các phần tử cho một tập hợp các kí tự: ');

Repeat

ch:=ReadKey;

tap:=tap+[ch];

Until not(ch in ['a'..'z']);

For ch:='a' to 'z' do

If ch in tap then Write(ch, ' ');

Readln;

END.

Hãy sửa lỗi và chạy thử chương trình.

17. Bằng cách tương tự như chương trình trên, bạn hãy lập chương trình không những có chức năng thêm phần tử vào tập hợp trực tiếp từ bàn phím mà còn có thể loại bớt phần tử khỏi tập hợp cũng trực tiếp từ bàn phím.

Bài 4. Mô phỏng đồ thị

A. Tóm tắt lý thuyết

1. Đồ thị vô hướng

Một đồ thị vô hướng G gồm hai tập hợp hữu hạn V và E , kí hiệu $G = (V, E)$; trong đó V là tập các đỉnh của đồ thị và E là tập các cạnh của đồ thị.

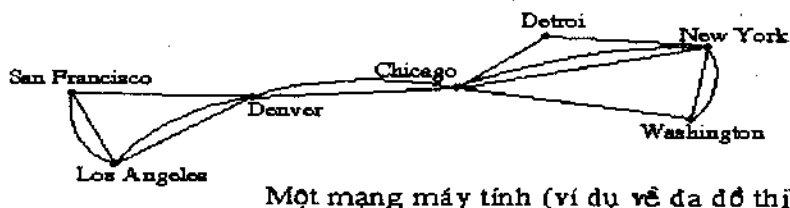
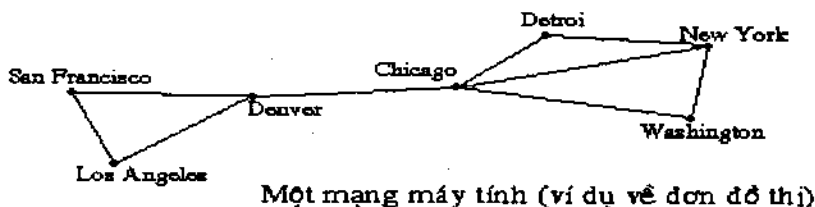
Giả sử ta có tập đỉnh V gồm n đỉnh là các số nguyên $\{1, 2, \dots, n\}$, khi đó E là tập hợp con các tích Descartes $V \times V$ mà nếu $(i, j) \in E$ thì $(j, i) \in E$, và ta đồng nhất (i, j) với (j, i) ; ($i, j = 1, 2, \dots, n$).

Nếu (i, j) là một cạnh của đồ thị ta nói cạnh đó được nối từ đỉnh i đến đỉnh j và hai đỉnh i và j là kề nhau.

Bậc của một đỉnh trong đồ thị là số đỉnh kề với nó.

Bậc của đồ thị là số lớn nhất trong số các bậc của tất cả các đỉnh của đồ thị đó.

Một đồ thị được gọi là đơn đồ thị nếu giữa hai đỉnh khác nhau chỉ có nhiều nhất 1 cạnh nối; nếu giữa hai đỉnh nào đó có nhiều hơn 1 cạnh nối thì đồ thị được gọi là đa đồ thị. Hình vẽ sau cho ví dụ về hai loại đồ thị.



Trong đồ thị phía trên: đỉnh Detroit kề với đỉnh Chicago và New York, Chicago kề với Denver, Detroit, New York và Washington... Đỉnh Detroit không kề với Denver vì giữa chúng không có cạnh nối, tuy nhiên chúng

liên thông với nhau vì giữa chúng có đường đi (qua Chicago hoặc qua New York và Chicago...).

Từ đây về sau chúng ta sẽ chỉ xét các đơn đồ thị.

2. Đồ thị có hướng

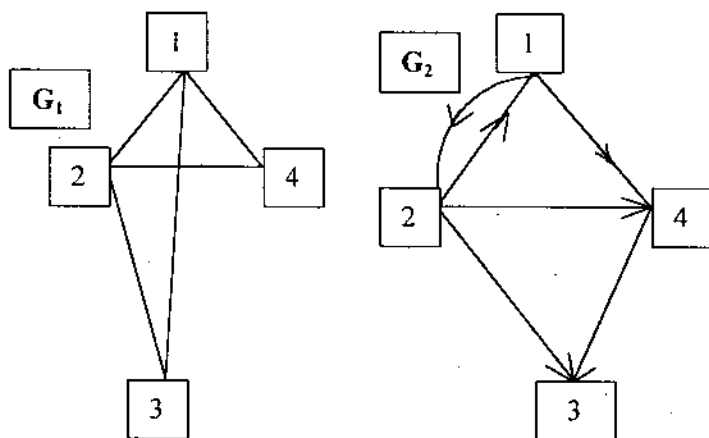
Đồ thị có hướng $G = (V, E)$; trong đó: V là tập các đỉnh của đồ thị; E là tập các cung của đồ thị.

Nếu (i, j) là một cung thì ta nói (i, j) là một cung đi từ đỉnh i tới đỉnh j .

Chú ý

Trong đồ thị vô hướng thì (i, j) là một cạnh cũng như là (j, i) . Nhưng trong đồ thị có hướng cung (i, j) hoàn toàn khác với cung (j, i) .

Ví dụ: Ta có đồ thị vô hướng G_1 và đồ thị có hướng G_2 dưới đây:



3. Biểu diễn đồ thị trên máy tính

+ Biểu diễn bằng ma trận kề

Cho đồ thị $G = (V, E)$ vô hướng (hoặc có hướng). Để ghi nhận các cạnh (hay cung) của G ta dùng một ma trận hai chiều $a[1..n, 1..n]$, n là số đỉnh của đồ thị, trong đó $a[i, j] = 1$ hay $a[i, j] = 0$ tùy thuộc vào có hay không có cạnh (hay cung) từ i đến j . Khi đó, ma trận a gọi là **ma trận kề** của đồ thị G .

Ví dụ, với đồ thị vô hướng G_1 và đồ thị có hướng G_2 ở trên ta có ma trận kề như sau:

$$G_1: \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \quad G_2: \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Nhận xét

Vì trong đồ thị vô hướng cạnh (i,j) cũng là cạnh (j,i) nên ta có $a[i,j] = a[j,i]$, do đó ma trận kề của đồ thị vô hướng là ma trận đối xứng.

+ **Biểu diễn bằng ma trận liên thuộc**

- Với đồ thị có hướng: $G = (V, E)$, tập V gồm n đỉnh $(1, 2, \dots, n)$ và tập E gồm m cung $(e_1, e_2, \dots, e_m)$. Ma trận liên thuộc để biểu diễn đồ thị G kí hiệu là $b[1..n, 1..m]$ được xác định như sau:

$$\begin{aligned} b[i,j] &= -1 \text{ nếu } i \text{ là đỉnh đầu của cung } e_j. \\ &= 1 \text{ nếu } i \text{ là đỉnh cuối của cung } e_j. \\ &= 0 \text{ nếu } i \text{ không phải là đỉnh kề của } e_j. \end{aligned}$$

- Với đồ thị vô hướng, $b[i,j]$ được xác định như sau:

$$\begin{aligned} b[i,j] &= 1 \text{ nếu } i \text{ là đỉnh đầu hay đỉnh cuối của cung } e_j. \\ &= 0 \text{ nếu } i \text{ không phải là đỉnh kề với } e_j. \end{aligned}$$

+ **Biểu diễn qua danh sách kề**

Với mỗi đỉnh i của đồ thị ta có một danh sách tất cả các đỉnh kề với i , ta kí hiệu là $List(i)$. Để thể hiện $List(i)$ ta có thể dùng các kiểu dữ liệu kiểu tập hợp hay kiểu mảng.

Ví dụ: với đồ thị G_1 ta có $List(1) = [2, 3, 4]$.

+ **Biểu diễn liệt kê các cạnh (cung)**

Đối với đồ thị vô hướng $G = (V, E)$ có tập E gồm m cạnh, ta dùng một mảng $a[1..m, 1..2]$ để liệt kê tất cả các cạnh của đồ thị. Cạnh i của đồ thị sẽ nối đỉnh $a[i, 1]$ với đỉnh $a[i, 2]$. Để tránh thiếu sót ta liệt kê từ đỉnh 1, 2, ... trở đi.

Ví dụ: với đồ thị G_1 ta liệt kê các cạnh:

1	2	2	4
1	3	3	1
1	4	3	2
2	1	4	1
2	3	4	2

Đối với đồ thị có hướng ta liệt kê tương tự.

Chú ý: Đối với đồ thị vừa có cạnh (m1 cạnh) và vừa có cung (m2 cung) ta liệt kê m1 cạnh trước, sau đó mới liệt kê tới m2 cung.

4. Đường đi và tính liên thông

Định nghĩa

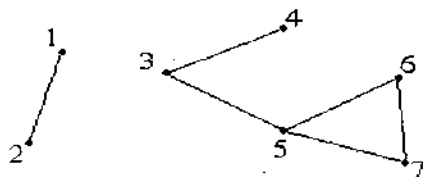
Một đường đi từ đỉnh u tới đỉnh v là một dãy cạnh $e_1, e_2, \dots, e_k$ mà $e_i = (u_i, u_{i+1}), 1 \leq i \leq k-1, u_1 = u; u_k = v$.

Một đường đi khép kín ($v = u$) gọi là một chu trình.

Đồ thị gọi là liên thông nếu với mọi cặp đỉnh (u, v) của nó ta luôn có đường đi từ u tới v . Từ đó có thể suy ra một đồ thị không liên thông nếu tồn tại một cặp đỉnh (u, v) mà không có đường đi nào nối chúng.

B. Bài tập

1. Bạn hãy cho một vài ví dụ về đồ thị: đồ thị vô hướng, đồ thị có hướng, đơn đồ thị, đa đồ thị
2. Trong một nhóm giao lưu sinh hoạt trại hè có 5 bạn: An, Bình, Công, Duy và Thanh. Các bạn An, Bình, Công đã quen nhau từ trước, Duy quen với Thanh. Công, Bình, còn Thanh quen với Duy và An. Bạn hãy vẽ đồ thị mô tả sự quen biết giữa các bạn trong nhóm.
3. Bạn hãy lập ma trận kề cho đồ thị ở Bài 2.
4. Hãy lập ma trận kề cho đồ thị sau:



Trong các bài tập từ nay về sau, ma trận kề của mỗi đồ thị được chứa trong một text file với qui cách như sau:

- Dòng đầu tiên chứa duy nhất một số nguyên n là số đỉnh của đồ thị.
- n dòng tiếp theo mỗi dòng chứa n số nguyên phân cách nhau bởi dấu cách, là các phần tử trên mỗi dòng của ma trận kề.

5. Một bàn cờ 8×8 được đánh số thứ tự các ô theo cách sau:

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Mỗi ô có thể coi là một đỉnh của đồ thị. Hai đỉnh được coi là kề nhau nếu một con vua đặt ở ô này có thể nhảy sang ô kia sau 1 nước đi. Ví dụ ô 1 kề với 2, 9 và 10, ô 11 kề với 2, 3, 4, 10, 12, 18, 19, 20... Hãy lập chương trình tạo ma trận kề của đồ thị, kết quả ghi ra text file P20405.TXT

6. Một bàn cờ 8×8 được đánh số thứ tự các ô như bài trên. Mỗi ô có thể coi là một đỉnh của đồ thị. Hai đỉnh được coi là kề nhau nếu một con mã đặt ở ô này có thể nhảy sang ô kia sau 1 nước đi. Ví dụ ô 1 kề với 11 và 18, ô 11 kề với 1, 5, 17, 21, 26, 28 ... Hãy lập chương trình tạo ma trận kề của đồ thị, kết quả ghi ra text file P20406.TXT.

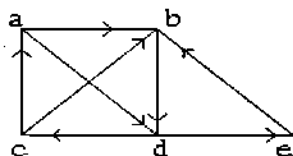
7. Cho trước ma trận kề của một đồ thị như sau:

0	1	0	0	1	1
1	0	0	1	1	0
0	0	0	1	0	1
0	1	1	0	1	1
1	1	0	1	0	0
1	0	1	1	0	0

Hãy vẽ đồ thị đó.

8. Bạn hãy cho một vài ví dụ về đồ thị có hướng.

9. Bạn hãy lập ma trận kề cho đồ thị có hướng sau đây:



10. Cho trước ma trận kề của một đồ thị. Bạn hãy lập chương trình tạo danh sách kề của đồ thị đó.

Kết quả đưa ra text file với qui cách như sau:

- Dòng đầu tiên chứa duy nhất một số nguyên n là số đỉnh của đồ thị.
- n dòng tiếp theo mỗi dòng chứa một số số nguyên phân cách nhau bởi dấu cách, là các phần tử của một tập hợp tương ứng với một đỉnh.

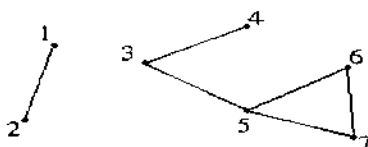
Trong các bài sau, danh sách kề được chứa trong file với qui cách như trên.

11. Cho trước danh sách kề của một đồ thị. Bạn hãy lập chương trình tạo ma trận kề của đồ thị đó.

12. Một trong những bài toán cơ bản nhất của lý thuyết đồ thị là bài toán tìm thành phần liên thông của đồ thị, nghĩa là phần cực đại của đồ thị mà giữa 2 điểm bất kì trong nó tồn tại ít nhất một đường đi. Để tìm thành phần liên thông chứa điểm s ta có thể làm theo cách sau:

- Trước tiên ta dùng một mảng Queue làm hàng đợi để chứa thành phần liên thông cần tìm, ghi s vào Queue (như vậy s sẽ được xét đầu tiên).
- Xét điểm s : Kiểm tra lần lượt tất cả các điểm trên đồ thị. Nếu điểm nào kề với s (tức là giữa nó và s có đường đi) thì ghi tiếp vào Queue.
- Xét điểm tiếp theo trong Queue, giả sử đó là điểm i : Kiểm tra lần lượt tất cả các điểm trên đồ thị. Nếu điểm nào kề với i (tức là giữa nó và i có đường đi) mà chưa có mặt trong Queue thì ghi tiếp vào Queue. Sau đó lại chuyển qua xét điểm tiếp theo...

Ta chỉ dừng lại khi nào đã xét hết các điểm trong Queue.



Để minh họa ta xét ví dụ tìm miền liên thông chứa đỉnh 3 trong đồ thị ở bài 4.

Các công việc phải làm thể hiện trong bảng sau:

Thứ tự	Tên công việc	Kết quả trong Queue
1	Khởi tạo Queue	{3}
2	Xét đỉnh 3	{3, 4, 5}
3	Xét đỉnh 4	{3, 4, 5}
4	Xét đỉnh 5	{3, 4, 5, 6, 7}
5	Xét đỉnh 6	{3, 4, 5, 6, 7}
6	Xét đỉnh 7	{3, 4, 5, 6, 7}

Trong Queue, đỉnh đã xét kí hiệu bằng chữ số in đậm. Kết quả cuối cùng ta được Queue chứa các đỉnh 3, 4, 5, 6, 7. Đó chính là thành phần liên thông chứa đỉnh 3.

Bây giờ cho trước ma trận kề của một đồ thị như sau:

0	1	0	0	0	1	1
1	0	0	1	0	0	0
0	0	0	1	0	0	1
0	1	1	0	0	0	0
0	0	0	0	0	0	1
1	0	0	0	0	0	1
1	0	1	0	1	1	0

Bạn hãy tìm thành phần liên thông chứa đỉnh thứ i , với $i = 1$ và $i = 5$. Kết quả thể hiện trong một bảng tương tự bài 12.

13. Cho trước ma trận kề của một đồ thị. Bạn hãy lập chương trình tìm thành phần liên thông chứa đỉnh s theo cách trên (s nhập từ bàn phím).

14. Một trong những bài toán cơ bản nữa của lý thuyết đồ thị là bài toán tìm đường đi. Để tìm một đường đi từ đỉnh s tới đỉnh t của đồ thị ta có thể làm theo cách được trình bày ở bài trên.

Ví dụ để tìm đường đi từ đỉnh 3 tới đỉnh 7 ta có thể làm như sau:

Khi đỉnh 7 đã được ghi vào Queue có nghĩa là đã có đường đi từ 3 tới 7 thì ta dừng lại. Tuy nhiên đường đi từ 3 tới 7 không phải là đường đi qua 3, 4, 5, 7 theo đúng thứ tự ta đã xét vì đường đi qua 4 là đường cụt. Để tránh các đường cụt, ta phải sử dụng một con số để đánh dấu các

đỉnh đã ghi trong Queue: con số đó chỉ rõ ta đã đi tới đỉnh hiện thời từ đỉnh nào

Thứ tự	Tên công việc	Kết quả trong Queue
1	Khởi tạo Queue	{3}
2	Xét đỉnh 3	{3, 4, 5} {3, 3}-đánh dấu
3	Xét đỉnh 4	{3, 4, 5} {3, 3}
4	Xét đỉnh 5	{3, 4, 5, 6, 7} {3, 3, 5, 5}-đánh dấu

Như vậy khi đã đi tới đích, ta có thể lần ngược lại con đường đã đi. Trong ví dụ trên, ta đã đi tới đỉnh 7 từ đỉnh 5, ta đã đi tới đỉnh 5 từ đỉnh 3. Như vậy con đường ta đã đi từ 3 tới 7 là đường đi qua 3, 5, 7.

Bây giờ cho trước ma trận kề của một đồ thị như sau:

```

0 1 0 0 0 1 1
1 0 0 1 0 0 0
0 0 0 1 0 0 0
0 1 1 0 0 0 0
0 0 0 0 0 0 1
1 0 0 0 0 0 1
1 0 1 0 1 1 0

```

Bạn hãy tìm một đường đi từ đỉnh 2 tới đỉnh 6. Kết quả thể hiện trong bảng.

15. Cho trước ma trận kề của một đồ thị. Bạn hãy lập chương trình tìm đường đi từ đỉnh s tới đỉnh t theo cách trên (s và t nhập từ bàn phím).

16. Bằng cách kết hợp chương trình đã lập ở Bài 15 và Bài 6, bạn hãy lập chương trình tìm một đường đi của con mã trên bàn cờ từ ô s tới ô t (s và t nhập từ bàn phím). Hãy kiểm tra kết quả trên thực tế.

Bài 5. Kiểu bản ghi

A. Tóm tắt lý thuyết

1. Khái niệm và định nghĩa

Để tạo ra một cấu trúc dữ liệu mới với các phần tử dữ liệu có các kiểu khác nhau, người ta định nghĩa ra bản ghi (Record). Bản ghi là một cấu trúc bao gồm nhiều thành phần. Các thành phần có thể thuộc các kiểu dữ liệu khác nhau và được gọi là các trường (Field), mỗi trường đều được đặt tên.

Ví dụ: Để mô tả kiểu R có cấu trúc bản ghi với danh sách các trường có tên là F1, F2, ..., Fn thuộc các kiểu dữ liệu tương ứng là K1, K2, ..., Kn ta dùng cách viết như sau:

```

Type
  R = Record
    F1 : K1;
    F2 : K2;
    ...
    Fn : Kn;
  End;

```

2. Cách sử dụng Record

Để truy nhập một biến kiểu Record, ta truy nhập theo thành phần của chúng qua cấu trúc sau:

<Tên biến Record>.<Tên trường>

3. Câu lệnh With

+ Cú pháp lệnh:

Câu lệnh With có thể lồng nhau.

With <biến kiểu Record> **Do** <câu lệnh>

B. Câu hỏi và bài tập

1. Tìm lỗi trong các khai báo sau:

```
Type hocsinh=record:
    hoten: string[26];
```

```

        tuoi: byte;
    end;
giaovien=array[1..10] of string[26];
lop=record
    ten: string[4];
    gv: giaovien;
    hs: hocsinh;
end;
Var class: lop

```

2. Cho chương trình:

```

Program Doc_Ghi_Record;
Uses Crt;
Var a: record
    hoten: string[26];
    tuoi: byte;
end;
Begin
    write('Nhap vao a:'); readln(a);
    writeln('Ghi tu a ra:'); write(a);
    readln;
End.

```

Hãy tìm lỗi trong chương trình trên và sửa lại cho đúng.

3. Cho chương trình nhập dữ liệu vào một mảng 20 bản ghi:

```

Program Mang_Record;
Uses Crt;
Type
    kieu=record
        hoten: string[26];
        tuoi: byte;
    end;
Var i: integer;
    a: array[1..20] of kieu;
Begin
    writeln('Nhap vao f. Thoi nhap khi i=20. ');
    for i:=1 to 20 do
        begin

```



```
write('Hoten: '); readln(a[i].hoten);  
write('Tuoi: '); readln(a[i].tuoi);
```

```
end;
```

End.

Hãy biến đổi chương trình sao cho:

- Các trường của mỗi bản ghi được nhập trên cùng một dòng.
- Dùng câu lệnh WITH để chương trình gọn hơn.

4. Trong trường hợp nào có thể thay thế mảng (một chiều) các bản ghi bằng một mảng 2 chiều?

5. Thông tin về mỗi học sinh gồm:

- Họ đệm: một xâu 25 ký tự.
- Tên: một xâu 10 ký tự.
- Tuổi: một số nguyên 2 chữ số.
- Lớp: một xâu 2 chữ số và một chữ cái viết hoa.

Hãy lập chương trình nhập từ bàn phím danh sách một lớp 15 học sinh vào một mảng bản ghi. Sau đó hiển thị danh sách lên màn hình, mỗi người một dòng.

6. Cho chương trình nhập dữ liệu vào một file bản ghi:

```
Program File_Record;  
Uses Crt;  
Type kieu=record  
    hoten: string[26];  
    tuoi: byte;  
end;  
Var a: kieu;  
    f: file of kieu;  
Begin  
    Assign(f,'recfile');  
    Rewrite(f);  
    Writeln('Nhap vao f. Thoi nhap khi hoten="" (go Enter).');  
    Write('hoten: '); Readln(a.hoten);  
    While (a.hoten<>") do  
        Write('tuoi: '); Readln(a.tuoi);  
        Write(f,a);  
        Write('hoten: '); Readln(a.hoten);
```

end;
Close(f);
End.

Hãy chạy thử chương trình trên và xem kết quả.

7. Thông tin về mỗi học sinh giống như bài trên. Hãy lập chương trình nhập danh sách một lớp 20 học sinh vào một file bản ghi.

8. Thông tin về mỗi học sinh là một bản ghi gồm các trường:

- Hodem: một xâu 25 kí tự.
- Ten: một xâu 10 kí tự.
- Tuoi: một số nguyên 2 chữ số.
- Lop: một xâu 2 chữ số và một chữ cái viết hoa.

Một file bản ghi chứa một danh sách một lớp gồm 20 học sinh. Hãy lập chương trình hiển thị danh sách lên màn hình, mỗi người một dòng.

9. Một file bản ghi chứa một danh sách một lớp gồm 20 học sinh, thông tin về mỗi học sinh giống như bài 8. Hãy lập chương trình hiển thị danh sách đã sắp xếp lên màn hình, mỗi người một dòng. Yêu cầu sắp xếp theo tên, nếu trùng tên thì sắp xếp theo họ đệm. Cho phép dùng một mảng bản ghi trung gian để sắp xếp.

10. Một file bản ghi chứa một danh sách một lớp gồm 20 học sinh, thông tin về mỗi học sinh giống như bài 8. Hãy lập chương trình sắp xếp lại danh sách học sinh trong file. Cho phép dùng một mảng bản ghi trung gian để sắp xếp.

11. Một file bản ghi chứa một danh sách học sinh, thông tin về mỗi học sinh giống như bài 8. Hãy lập chương trình hiển thị danh sách học sinh lên màn hình. Không cho phép dùng mảng trung gian.

12. Một file bản ghi chứa một danh sách học sinh PTTH, thông tin về mỗi học sinh giống như bài 8. Hãy lập chương trình tách danh sách ra thành 3 file, mỗi file chứa học sinh của một khối.

13. Một file bản ghi chứa một danh sách học sinh, thông tin về mỗi học sinh giống như bài 8. Hãy lập chương trình tạo một file bản ghi khác chứa danh sách đó, mỗi bản ghi gồm các trường:

- Hoten: một xâu 35 kí tự.
- Tuoi: một số nguyên 2 chữ số.
- Khoi: một số nguyên 2 chữ số.

- Lop: một chữ cái viết hoa.

14. Một file bản ghi chứa một danh sách học sinh PTTH, thông tin về mỗi học sinh ngoài các trường Hodem, Ten, Tuoi, Lop giống như các bài trên còn có thêm trường Diem chứa điểm trung bình của học sinh trong năm học. Hãy lập chương trình:

- Hiện thị lên màn hình danh sách những học sinh giỏi nhất của trường là những bạn có điểm trung bình từ 8.0 trở lên và cao nhất trong khối.
- Lập danh sách học sinh trong năm học mới, biết một học sinh có điểm trung bình từ 5.0 trở lên thì được lên lớp. Chú ý: lớp 10A lên lớp 11A, lớp 11A lên lớp 12A... Kết-quả chứa trong file.

15. Cho file bản ghi f chứa dữ liệu về kho sách, dữ liệu về mỗi cuốn sách được chứa trong một bản ghi gồm 3 trường mang thông tin về:

- Họ tên tác giả: một xâu 26 kí tự.
- Tên sách: một xâu 40 kí tự.
- Năm xuất bản: một số nguyên 4 chữ số.

Hãy lập chương trình nhập dữ liệu vào kho sách, sau đó tìm ra:

- Những cuốn sách của một tác giả cho trước xuất bản vào một năm cho trước.
- Những cuốn sách có tên cho trước.

Kết quả thể hiện trên màn hình.

16. File bản ghi f chứa danh sách các ngày lễ trong một năm, mỗi bản ghi gồm ngày tháng, tên ngày lễ và số ngày được nghỉ. Hãy lập chương trình nhập danh sách các ngày lễ và tính:

- Tổng số các ngày lễ và tổng số các ngày nghỉ lễ trong năm.
- Tổng số các ngày lễ và tổng số các ngày nghỉ lễ trong quý 1, quý 2,...

Kết quả thể hiện trên màn hình.

Bài 6. Kiểu file

A. Tóm tắt lý thuyết

1. Các thao tác trên tệp

+ Mở tệp mới để lưu dữ liệu:

Assign(biến_tệp, Tên_tệp);

ReWrite(biến_tệp);

Ghi dữ liệu vào tệp thông qua thủ tục Write:

Write(biến_tệp, biến1, biến2, ..., biếnN);

Trong đó: biến1, biến2, ..., biếnN là các giá trị cần ghi vào tệp, có thể là các hằng, biến, biểu thức.

+ Đọc dữ liệu từ một tệp đã có:

Assign(biến_tệp, tên_tệp);

Reset(biến_tệp);

Đọc dữ liệu từ một tệp thông qua thủ tục Read:

Read(biến_tệp, biến1, biến2, ..., biếnN);

Trong đó: biến1, biến2, ..., biếnN là các giá trị cần đọc từ tệp, có thể là các hằng, biến, biểu thức.

+ Đóng tệp: Sau khi đọc dữ liệu từ một tệp xong hay cất dữ liệu mới vào một tệp nên dùng thủ tục đóng tệp để đảm bảo thông tin trên tệp là đầy đủ và tin cậy, thủ tục đóng tệp có cú pháp:

Close(biến_tệp);

+ Có thể truy nhập trực tiếp vào một phần tử của tệp thông qua thủ tục SEEK, có cú pháp như sau:

SEEK(biến_tệp, Stt);

Trong đó: Stt là số thứ tự của phần tử trong tệp (phần tử đầu tiên của tệp đánh số là 0). Gặp lệnh này máy sẽ đặt con trỏ tệp vào phần tử thứ Stt để ta thực hiện các thao tác cần thiết ở vị trí đó.

2. Các hàm và thủ tục xử lý tệp

+ **Erase**(biến_tệp): xoá tệp trên đĩa có tên ấn định với biến_tệp. Chú ý là không được xoá tệp đang mở.

+ **Rename**(biến_tệp, S): dùng thay đổi tên tệp với tên mới kiểu String chứa trong chuỗi S. Tên mới không được trùng với các tên tệp nào đó đã có sẵn trên đĩa đang làm việc. Không được đổi tên tệp đang mở.

+ **FileSize(biến_tệp)**: cho số phần tử của tệp biến_tệp. Hàm nhận giá trị bằng 0 khi tệp rỗng.

+ **FilePos(biến_tệp)**: cho biết vị trí hiện tại của con trỏ tệp (phần tử đầu tiên của tệp là phần tử số 0). Muốn ghi thêm một phần tử vào cuối tệp trước hết ta phải đưa con trỏ tệp về vị trí cuối cùng của tệp qua lệnh sau:

SEEK(biến_tệp, FileSize(biến_tệp));

3. Tệp văn bản (Text Files)

- Khai báo tệp văn bản: biến tệp f có kiểu TEXT được khai báo như sau:
Var f:TEXT;

Thành phần cơ sở của tệp kiểu TEXT là kí tự. Tệp văn bản được cấu trúc theo dòng, mỗi dòng được kết thúc bởi dấu EOLn (End Of Line), với Turbo Pascal đó là các cặp kí tự điều khiển CR (Carriage Return: nhảy về đầu dòng, mã ASCII = 13) và LF (Line Feed: nhảy xuống dòng tiếp theo, mã ASCII = 10). Tệp được kết thúc bởi dấu EOF (End Of File), với Turbo Pascal đó là Ctrl-Z (^Z), mã ASCII = 26.

Các thủ tục đã trình bày đối với tệp cấu trúc cũng sử dụng được đối với tệp văn bản, tuy nhiên tệp văn bản cũng có những đặc điểm khác sau:

▪ Các hàm chuẩn với tệp văn bản:

EOF(var f:Text):Boolean;

Hàm này trả về giá trị True khi duyệt đến cuối tệp, ngược lại trả lại giá trị False. Ta thường dùng hàm này để kiểm tra xem đã đọc hết tệp văn bản chưa.

EOLN(var f:Text):Boolean;

Hàm này trả về giá trị False nếu duyệt chưa đến cuối dòng, ngược lại cho giá trị True. Hàm này thường dùng để kiểm tra xem đã đọc hết dòng chưa.

▪ Các thủ tục với tệp văn bản:

+ **Đọc dữ liệu từ tệp văn bản đã có trên đĩa:**

Read(biến_tệp, biến 1, biến 2, ..., biến N);

Readln(biến_tệp, biến 1, biến 2, ..., biến N);

Readln(biến_tệp);

Trong đó: biến 1, biến 2, ..., biến N có thể là các biến thuộc kiểu kí tự, nguyên, thực, logic, chuỗi. Lệnh Read đọc một hay nhiều phần tử mà không chuyển con trỏ tệp xuống dòng, Readln cũng đọc như Read nhưng lại chuyển con trỏ tệp xuống đầu dòng tiếp theo sau khi đã lần lượt đọc các biến tương ứng. Lệnh Readln(biến_tệp) chỉ đưa con trỏ tệp xuống đầu dòng tiếp theo mà không đọc gì cả.

+ Ghi vào một tệp văn bản:

Dữ liệu được ghi vào tệp thông qua các lệnh Write, Writeln, cụ thể như sau:

(1) **Write**(biến_tệp, biến 1, biến 2, biến n);

(2) **Writeln**(biến_tệp, biến 1, biến 2, biến n);

(3) **Writeln**(biến_tệp);

Lệnh (1) có tác dụng viết các giá trị của các biến 1, biến 2, biến n (các giá trị này là các hằng hay biểu thức có kiểu đơn giản như: nguyên, thực, chuỗi, logic, kí tự) vào biến tệp.

Lệnh (2) tương tự như lệnh (1) nhưng đưa dấu hiệu hết dòng vào tệp sau khi đã ghi hết các giá trị đó.

Lệnh (3): chỉ thực hiện việc đưa thêm dấu hiệu hết dòng vào tệp.

+ Thủ tục thêm dòng vào tệp:

APPEND(var f:Text);

Lệnh Append mở tệp văn bản ra để bổ sung thêm các dòng, định vị con trỏ tệp vào cuối tệp.

B. Câu hỏi và bài tập

1. Các khai báo sau đây đúng hay sai:

Type

e = file of byte;

f = file of string[10];

g = file of array[1..100] of real;

h = file of array[1..100, 1..100] of real;

i = file of set of char;

2. Khai báo sau đây đúng hay sai:

Type

a = array[1..10] of integer;

b = record

 hoten: string[26];

 tuoi: byte;

end;

Var

f: file of a;

g: file of b;

3. Hai khai báo sau có gì khác nhau:

Type a = array[1..10] of integer;

Var f: file of a;

Và

Var f: file of array[1..10] of integer;

4. Chương trình sau đây minh hoạ việc khai báo, tạo mới và ghi vào một tệp các xâu:

```
Program Ghi_file_moi;
Uses Crt;
Type
    kieu=string[10];
Var i: integer;
    a: kieu;
    f: file of kieu;
BEGIN
    Assign(f,'Ghi_Doc'); Rewrite(f);
    Writeln('Nhập vào f.Thời nhập khi f="" (go Enter).');
    i:=0;
    Write('f.i, ' '); Readln(a);
    While (a<>"") do
        begin
            Write(f,a);
            Inc(i);
            Write('f.i, ' '); Readln(a);
        end;
    Close(f);
END.
```

Hãy chạy thử chương trình trên.

5. Hãy lập chương trình tạo một tệp chứa các số nguyên nhỏ hơn 1000.

6. Hãy lập chương trình tạo một tệp số nguyên chứa các số nguyên tổ nhỏ hơn 10000 theo thứ tự tăng dần.

7. Cho f là tệp văn bản chứa các xâu 10 kí tự. Hãy lập chương trình nhập và hiển thị nội dung file đó lên màn hình, mỗi xâu một dòng, đầy trang màn hình thì dừng lại đợi gõ Enter mới hiển thị trang tiếp cho đến hết.

8. Chương trình sau đây minh hoạ việc khai báo và đọc một tệp các xâu:

```
Program Doc_file;
Uses Crt;
```

```

Type kieu=string[10];
Var i: integer; a: kieu; f: file of kieu;
Begin
  assign(f,'Ghi_Doc'); reset(f);
  writeln('Doc ra tu f. Thoi doc khi Eof(f).');
  i:=0;
  while not Eof(f) do
    begin
      read(f,a);
      writeln('f,i.' = ',a);
      inc(i);
    end;
  readln;
  close(f);
End.

```

Hãy chạy thử chương trình trên. So sánh với bài 4.

9. Chương trình sau đây minh hoạ việc khai báo và ghi thêm vào cuối một tệp các xâu:

```

Program Ghi_them_vao_cuoi_file;
Uses Crt;
Type kieu=string[10];
Var i: integer;
    a: kieu;
    f: file of kieu;
Begin
  assign(f,'Ghi_Doc');
  reset(f);
  writeln('Ghi them vao cuoi file f. Thoi khi fi=""');
  i:=FileSize(f);
  seek(f,i);
  write('f,i.' = '); readln(a);
  while (a<>"") do
    begin
      write(f,a);
      inc(i);
      write('f,i.' = '); readln(a);
    end;
  close(f);
End.

```



```
end;  
close(f);  
End.
```

Hãy chạy thử chương trình đó. So sánh với bài 7.

10. Viết chương trình cho phép đọc dữ liệu từ bàn phím và ghi thêm vào cuối một tệp các bản ghi.

11. Sửa lại chương trình trên sao cho ta có thể ghi đè lên (sửa) một bản ghi đã có với thứ tự của bản ghi đó do ta tùy chọn.

12. Bạn hãy lập chương trình cho phép loại (xoá) một bản ghi với thứ tự tùy ý ra khỏi file.

13. Lập chương trình cho phép chèn một bản ghi vào file ở vị trí tùy ý.

14. Cho file bản ghi với 2 trường Tên, Tuổi. Hãy lập chương trình tạo ra một file chỉ mục tương ứng là một file số nguyên chỉ rõ thứ tự của bản ghi trong danh sách đã được sắp xếp theo Tên. Giả định rằng file không dài lắm (ít hơn 500 bản ghi) và được dùng một mảng trung gian để sắp xếp.

15. Kết hợp các chương trình ở các bài trên (10, 11, 12, 13) để được một chương trình mới cho phép xem và tùy ý sửa chữa, xoá các bản ghi có sẵn hoặc thêm bản ghi mới vào một file bản ghi. Hãy lập chương trình thực hiện các thao tác trên với file bản ghi có 2 trường:

- Ten: string[30];
- Tuoi: byte;

Chương trình phải hiển thị một Menu với các tùy chọn Xem danh sách, Sửa, Xoá, Thêm một bản ghi và Thoát để thực hiện các chức năng tương ứng.

16. Cho một file bản ghi và một file chỉ mục tương ứng như bài 14. Hãy hiển thị danh sách đã được sắp xếp lên màn hình với sự trợ giúp của file chỉ mục.

Bài 7. Kiểu con trỏ

A. Tóm tắt lý thuyết

1. Biến tĩnh và biến động

Các biến thuộc kiểu dữ liệu mà ta đã biết như kiểu mảng, tập hợp, bản ghi ... được gọi là biến tĩnh vì chúng được xác định rõ ràng khi khai báo và sau đó được dùng thông qua tên của chúng. Thời gian tồn tại của các biến tĩnh cũng là thời gian tồn tại của khối chương trình có chứa khai báo các biến này. Do đó trong nhiều chương trình, việc sử dụng một số lượng lớn các biến sẽ gây hậu quả là không đủ bộ nhớ để lưu trữ các biến. Để khắc phục tình trạng lãng phí bộ nhớ, Turbo Pascal cho phép dùng biến động. Các biến động được lưu giữ trong vùng Heap, khi cần có thể tạo ra để chứa dữ liệu, khi không cần có thể xóa đi để giải phóng bộ nhớ. Do biến này hoàn toàn không được xác định từ trước, nó không có tên và được quản lý bởi con trỏ.

2. Biến con trỏ

Biến con trỏ (Pointer variable) là một loại biến đặc biệt, có kích thước 2 byte, không dùng để chứa dữ liệu mà chứa địa chỉ của biến động.

+ Khai báo: Kiểu con trỏ TP và biến con trỏ P tương ứng được dùng để chứa địa chỉ của các biến động có kiểu T được định nghĩa như sau:

Type TP = ^T; Var P:TP;

Để thâm nhập vào biến động có địa chỉ nằm trong biến con trỏ P ta dùng kí hiệu: P[^].

• Các thao tác với con trỏ

+ Phép gán:

P:=Q; {Q trỏ đến vùng nhớ nào thì P cũng trỏ tới vùng nhớ đó}

+ So sánh hai con trỏ cùng kiểu: Với dữ liệu kiểu con trỏ không thể thực hiện được các phép so sánh như: <=, >= mà chỉ có thể so sánh bằng nhau (=) và khác nhau (<>).

+ Hằng con trỏ NIL: là một giá trị hằng đặc biệt dành cho biến con trỏ để báo con trỏ không trỏ vào đâu cả. Ta có thể gán NIL cho bất kì biến con trỏ nào.

3. Các thủ tục với kiểu con trỏ

+ Cấp phát vùng nhớ cho biến động do con trỏ P trỏ tới:

New(P);

+ Giải phóng ô nhớ của một biến động:

Dispose(P)

+ Để giải toả vùng Heap ta dùng hai thủ tục Mark và Release:

Mark(PVar); sẽ gán giá trị của con trỏ Heap cho PVar (PVar là một con trỏ dùng để đánh dấu vị trí đầu của vùng ô nhớ cần giải phóng).

Release(PVar); xoá tất cả các biến động đã tạo từ khi đánh dấu bởi Mark.

B. Câu hỏi và bài tập

1. Đoạn chương trình sau mô tả việc khai báo kiểu con trỏ và biến kiểu con trỏ:

Program P20701;

Type

BytePtr = ^Byte;

WordPtr = ^Word;

IdentPtr = ^IdentRec;

IdentRec = record

Ident: string[15];

Next: IdentPtr;

end;

Var

a: BytePtr;

b: WordPtr;

c: IdentPtr;

d: ^String[10];

Bạn hãy tìm chỗ sai và sửa lỗi cho đoạn chương trình trên.

2. Đoạn chương trình sau mô tả việc gán trị cho một con trỏ và lấy giá trị tại vùng nhớ mà nó trỏ tới:

Program P20702;

Type

kdl=integer;

Var i: ^kdl;

```

a: kdl;
BEGIN
i := Addr(a); {gán địa chỉ của a cho i, có thể viết i:=@a }
a:=100;
Write(i^);
END.

```

Bạn hãy thử chạy chương trình trên.

Trong chương trình trên nếu lệnh viết ra màn hình được viết như sau:

```
Write(i);
```

thì ta sẽ nhận được kết quả gì? Vì sao?

3. Theo mẫu chương trình trên, bạn hãy lập chương trình nhập giá trị cho nhiều biến có kiểu khác nhau, sau đó truy cập vào địa chỉ của chúng để lấy giá trị và viết ra màn hình.

4. Cho đoạn chương trình sau:

```

Type
ref=^integer;
Var p, q: ref;
Begin
P^:=5; q^:=6; p^:=q^;
If p=q then p:=nil else if p^=q^ then q:=p;
If p=q then q^:=4;
Writeln(p^);
End.

```

a. Hỏi kiểu của p, của p^ là gì? Giá trị của p, của p^ bằng bao nhiêu?

b. Kết quả in ra là gì?

5. Bạn hãy chạy thử chương trình sau:

```

Program P20705;
Type
kdl=integer;
Var i: ^kdl;
a: kdl;
BEGIN
i:=Addr(a);
a:=100;
Writeln(i^);

```

```

i^:=200;
Writeln(a);
Readln;
END.

```

Rõ ràng trong chương trình trên ta đã sử dụng $i^$ như là biến a. Nếu thay dòng lệnh $i:=\text{Addr}(a)$; bằng dòng lệnh $a:=i^$; thì ta có được kết quả như cũ không? Vì sao?

6. Đoạn chương trình sau mô tả việc tạo một biến con trỏ mới khi chạy chương trình:

```

Program P20706;
Type
  Str18=string[18];
Var
  P: ^Str18;
BEGIN
  New(P);
  P^:='Now you see it...';
  Dispose(P); { Now you don't... }
END.

```

Hãy lưu ý rằng biến đó không có tên nhưng ta vẫn có thể làm việc với nó thông qua P.

Bạn hãy tìm chỗ sai, sửa lỗi cho đoạn chương trình trên và chạy thử nó.

7. Đoạn chương trình sau mô tả việc khai báo một vùng nhớ để sử dụng:

```

Program P20707;
Type
  Rec=record
    Name: string[20];
    Age : Byte;
  end;
Var
  i: byte;
  p: pointer;
BEGIN
  i:=SizeOf(Rec); {tính kích thước một bản ghi }
  GetMem(p,i); {định vị một vùng nhớ; địa chỉ, kích thước :
  { ...}

```

```

{ ...sử dụng vùng nhớ... }
{ ...}
FreeMem(p.i); {giải phóng vùng nhớ }

```

END.

Bạn hãy chạy thử chương trình và tham khảo thêm bài 16.

8. *Danh sách được ghép nối* là một danh sách mà trong mỗi record, ngoài trường chứa dữ liệu còn có một trường Next chứa địa chỉ của record tiếp sau. Nó có thể tạo nên bởi chương trình sau:

```

Program P20708;
Type
    Ptr=^Rec;
    Rec=record
        Name: string[20];
        Next: Ptr;
    end;
Var p: Ptr;
BEGIN
    New(p);
    Write('Ten: '); readln(p^.name);
    Repeat
        New(p^.next);
        p:=p^.next;
        Write('Ten: '); readln(p^.name);
    Until p^.name="";
END.

```

Chương trình trên cho phép ta nhập một danh sách Tên được ghép nối từ bàn phím. Bạn hãy chạy thử chương trình và bổ sung cho nó chức năng liệt kê danh sách đã nhập.

9. Trong bài trên, trường Next trong mỗi record cho phép ta lần tới record tiếp sau nó, tức là ta chỉ có thể duyệt danh sách theo một chiều. Để từ mỗi record có thể lần tới record đứng trước nó, ta phải dùng thêm một trường chứa địa chỉ của record đứng trước nó. Ví dụ:

```

Type
    Ptr = ^Rec;
    Rec = record
        Previous: Ptr;

```

```
Name: string[20];  
Next: Ptr;  
end;
```

...

Bạn hãy lập chương trình tạo nên một danh sách mà mỗi record được ghép nối với cả record đứng trước lẫn record đứng sau nó. Chương trình cho phép ta nhập một danh sách Tên từ bàn phím, sau đó liệt kê danh sách đã nhập theo cả hai thứ tự xuôi và ngược.

10. Bạn hãy lập chương trình cho phép ta nhập một danh sách Tên từ một text file, tạo nên một danh sách mà mỗi record được ghép nối với cả record đứng trước lẫn record đứng sau nó. Chương trình sẽ liệt kê danh sách đã nhập theo cả hai thứ tự xuôi và ngược.

11. Bạn hãy lập chương trình cho phép nhập từ bàn phím một danh sách được ghép nối. Sau đó gỡ bỏ một record khỏi danh sách.

12. Một lớp học gồm n bạn xếp thành vòng tròn. Bắt đầu từ bạn thứ 1 ta đếm theo chiều kim đồng hồ, đến bạn thứ k sẽ bị loại ra khỏi vòng tròn; sau đó lại bắt đầu đếm từ bạn kế tiếp. Bạn hãy lập chương trình cho phép nhập một danh sách n học sinh theo thứ tự xếp trên vòng tròn, sau đó liệt kê lại danh sách theo thứ tự bị loại khỏi vòng. Danh sách học sinh ban đầu ghi trong tệp IN.TXT, kết quả đưa ra tệp OUT.TXT. Số k được nhập từ bàn phím.

13. Bạn hãy lập chương trình cho phép nhập một danh sách được ghép nối. Sau đó chèn thêm một record vào danh sách.

14. Bạn hãy lập chương trình cho phép nhập một danh sách được ghép nối. Sau đó đổi chỗ hai record trong danh sách.

15. Bạn hãy lập chương trình cho phép nhập một danh sách Tên được ghép nối. Sau đó sắp xếp lại danh sách theo trường Tên.

16. Chương trình sau thực hiện việc lưu ảnh một vùng màn hình hình chữ nhật ở tọa độ (10, 20, 30, 40) vào một biến bộ nhớ kiểu con trỏ rồi hiển thị ra ở một vị trí khác:

```
Program P20716;  
Uses Crt, Graph;  
Var Gd, Gm: Integer;  
P: Pointer;
```

```

        Size: Word;
BEGIN
Gd:=Detect;
InitGraph(Gd,Gm,' ');
Bar(0,0,GetMaxX div 2,GetMaxY);
Size:=ImageSize(10, 20, 30, 40); {tính kích thước ảnh}
GetMem(P,Size); {định vị một vùng nhớ: địa chỉ, kích thước}
GetImage(10,20,30,40,P^); {lưu ảnh màn hình vào bộ nhớ}
Readln; {chờ gõ Enter}
PutImage(500,100,P^,NormalPut); {đặt ảnh ra màn hình}
Readln;
CloseGraph;
END.

```

Hãy chạy thử chương trình.

17. Bạn hãy lập chương trình vẽ hình sau, sau đó lưu hình vào một biến bộ nhớ kiểu con trỏ rồi hiển thị ra ở một vị trí tùy chọn khác:



Bài 8. Vẽ đồ thị hàm số

Bài tập

Vẽ hệ trục tọa độ và đồ thị của các hàm số sau đây theo phương pháp vẽ từng điểm:

$$1. y = x/2.$$

$$2. y = x+1.$$

$$3. y = x^2/2.$$

$$4. y = x^2-3x+2.$$

$$5. y = \sin x$$

$$6. y = \cos x.$$

$$7. y = 1/x.$$

$$8. y = (x-1)/(x+1).$$

$$9. y = (x^2-x+1)/(x-1).$$

$$10. y = x^3/3-4x^2+2.$$

Các đường sau đây có phương trình cho dưới dạng tham số với tham số t , các hằng số được nhập trước từ bàn phím.

11. Đường thẳng đi qua (x_0, y_0) với hệ số góc b/a :

$$x = at+x_0$$

$$y = bt+y_0$$

với a, b, x_0, y_0 cho trước.

12. Đường tròn tâm (a, b) , bán kính r :

$$x = r.\cos t+a$$

$$y = r.\sin t+b$$

với r, a, b cho trước, $0 < t < 2\pi$.

13. Elip tâm (x_0, y_0) , trục lớn a , trục bé b :

$$x = a.\cos t+x_0$$

$$y = b.\sin t+y_0$$

với a, b, x_0, y_0 cho trước, $0 < t < 2\pi$.

14. Cho trước $(x_1, y_1), (x_2, y_2)$ là tọa độ của 2 điểm A và B. Hãy dựng đoạn thẳng AB.

15. Cho trước $(x_1, y_1), (x_2, y_2)$ là tọa độ của 2 điểm A và B. Hãy dựng hình chữ nhật ACBD với các cạnh song song với các cạnh màn hình.

16. Cho trước $(x_1, y_1), (x_2, y_2)$ là tọa độ của 2 điểm A và B và một số thực dương k . Hãy dựng điểm C thuộc AB sao cho C chia AB theo tỉ số k , nghĩa là $AC/CB = k$.

17. Vẽ đường cong Pascal:

$$x = a \cos^2 t + b \cos t$$

$$y = a \cos t \sin t + b \sin t$$

với $a, b > 0$ cho trước, $0 \leq t < 2\pi$.

Chú ý: Xét riêng các trường hợp $b \geq 2a, a < b < 2a, a > b$.

18. Vẽ đường cong Cardioid:

$$x = a \cos t (1 + \cos t)$$

$$y = a \sin t (1 + \cos t)$$

với $a > 0$ cho trước, $0 \leq t < 2\pi$.

19. Vẽ đường cong Astroid:

$$x = b \cos^3 t$$

$$y = b \sin^3 t$$

với $b > 0$ cho trước, $0 \leq t < 2\pi$.

20. Vẽ đường cong Cycloid:

$$x = at^2/(1+t^2)$$

$$y = at^3/(1+t^2)$$

Ở đây $a > 0$ cho trước, t nằm trong khoảng $(-\infty, +\infty)$.

21. Vẽ đường cong Strofoïd:

$$x = a(t^2-1)/(t^2+1)$$

$$y = at(t^2-1)/(t^2+1)$$

Ở đây $a > 0$ cho trước, t nằm trong khoảng $(-\infty, +\infty)$.

22. Vẽ đường cong Conhoid Nicomed:

$$x = a + l \cos t$$

$$y = at \sin t + l \sin t$$

với $a, l > 0$ cho trước.

Nhánh trái: $-\frac{\pi}{2} < t < \frac{\pi}{2}$.

Nhánh phải: $\frac{\pi}{2} < t < \frac{3\pi}{2}$.

Chú ý: Xét riêng các trường hợp: $l < a, l > a, l = a$.

Bài 9. Các thuật toán tạo chuyển động trên đồ họa

A. Tóm tắt lý thuyết

** Các hàm và thủ tục xử lý đồ họa*

+ **ImageSize**(x1,y1,x2,y2 : Integer); trả về số byte cần thiết để lưu trữ màn hình đồ họa giới hạn bởi hình chữ nhật (x1,y1,x2,y2).

+ **GetMem**(Var P : Point ; i : Integer); thủ tục tạo vùng nhớ (i bytes) quản lý bởi biến con trỏ.

+ **GetImage**(x1,y1,x2,y2 : Integer; Var Bitmap); chép các điểm nằm trong vùng màn hình giới hạn bởi (x1,y1,x2,y2) vào vùng nhớ có địa chỉ lưu trong biến bitmap (thường dùng biến động và được xác định kích thước bởi hàm ImageSize).

+ **PutImage**(x,y : Integer; Var Bitmap; CopyMode : word); chép các ảnh lưu trữ trong biến bitmap ra màn hình tại tọa độ (x,y); tham số CopyMode quy định cách thức chép ảnh: kết hợp ảnh đã có và ảnh được chép theo các phép toán logic (and, or, not, xor). Các hằng được định nghĩa trong Unit Graph:

CopyPut = 0	AndPut = 3
XorPut = 1	NotPut = 4
OrPut = 2	

B. Câu hỏi và bài tập

1. Chương trình sau hiển thị một hình vuông chuyển động theo chiều ngang trên màn hình từ trái qua phải với vận tốc đều:

```
Program P20901;  
Uses Crt, Graph;  
Var Gd, Gm, x, y: Integer;  
    Pa, Pb: Pointer;  
    Size: Word;  
BEGIN  
    Gd:=Detect;  
    InitGraph(Gd, Gm, '');  
    Size:=ImageSize(0, 0, 10, 10);  
    GetMem(Pb, Size);  
    GetImage(0, 0, 10, 10, Pb^); {hình vuông màu đen}
```

```

GetMem(Pa, Size);
Bar(0, 0, 10, 10);
GetImage(0, 0, 10, 10, Pa^); {hình vuông màu trắng}
ClearDevice;
x:=0; y:=200;
Repeat
PutImage(x, y, Pa^, NormalPut);
Delay(20);
PutImage(x, y, Pb^, NormalPut);
Inc(x);
If x>600 then x:=0;
Until KeyPressed;
CloseGraph;
END.

```

Hãy chạy thử chương trình.

Bạn hãy chú ý rằng trong chương trình, lệnh PutImage(x,y,Pb^,NormalPut) dùng để xoá hình vuông đã vẽ vì hình vuông chứa trong Pb^ cùng màu với màu nền.

2. Chương trình sau có chức năng giống hết chương trình ở bài trên:

```

Program P20902;
Uses Crt, Graph;
Var Gd, Gm, x, y: Integer;
    P: Pointer;
    Size: Word;
BEGIN
Gd:=Detect;
InitGraph(Gd, Gm, '');
Bar(1, 0, 11, 10);
Size:=ImageSize(0, 0, 11, 10);
GetMem(P, Size);
GetImage(0, 0, 11, 10, P^);
ClearDevice;
x:=0; y:=200;
Repeat
PutImage(x, y, P^, NormalPut);
Delay(20);

```

```

Inc(x);
If x>600 then x:=0;
Until KeyPressed;
CloseGraph;
END.

```

Hãy chạy thử chương trình.

Bạn hãy so sánh với chương trình ở bài trên và hãy giải thích: Vì sao ta không cần một câu lệnh riêng biệt để xóa hình vuông đã vẽ ?

3. Làm lại bài 1 với điều kiện: Vận tốc của chuyển động điều khiển được bằng bàn phím (chuyển động nhanh lên nếu gõ phím $\uparrow$, chậm đi nếu gõ phím $\downarrow$).
4. Hiển thị một hình chữ nhật trên màn hình, vị trí có thể điều khiển được bằng bàn phím: Gõ các phím mũi tên để dịch chuyển hình đó theo các hướng tương ứng.
5. Hãy hiển thị một điểm chuyển động xiên trên màn hình với vận tốc đều.
6. Hiển thị một điểm chuyển động đều theo chiều kim đồng hồ trên quỹ đạo tròn, tâm là tâm màn hình, bán kính $r = 150$.
7. Làm lại bài 6 với điều kiện: Vận tốc của chuyển động điều khiển được bằng bàn phím (chuyển động nhanh lên nếu gõ phím $\uparrow$, chậm đi nếu gõ phím $\downarrow$).
8. Làm lại bài 6 với chuyển động theo hướng ngược lại.
9. Vẽ trên màn hình một đoạn thẳng tự quay quanh một đầu cố định của mình.
10. Vẽ trên màn hình một đoạn thẳng tự quay quanh một điểm cố định của mình. Điểm đó chia đoạn thẳng theo tỉ số $k = -\frac{1}{3}$.
11. Vẽ trên màn hình một đoạn thẳng tự quay quanh một điểm cố định nằm ngoài đoạn thẳng đó.
12. Vẽ trên màn hình một hình vuông tự quay quanh tâm cố định của mình với vận tốc đều.

- 8.668
13. Vẽ một hình vuông đặc quay quanh tâm của mình với vận tốc đều.
 14. Vẽ một hình chữ nhật đặc quay quanh tâm của mình với vận tốc đều.
 15. Vẽ một hình vuông đặc quay quanh tâm của mình với vận tốc đều trong khi tâm hình vuông chuyển động thẳng (ngang hoặc xiên trên màn hình) với vận tốc đều.
 16. Hãy hiển thị một số điểm chuyển động theo chiều ngang trên màn hình từ trái qua phải với vận tốc khác nhau.

Bài 10. Các bài toán hình học

Bài tập

Trong các bài tập từ 1 đến 5, dữ liệu được nhập từ bàn phím, kết quả thể hiện trên màn hình.

1. Cho 4 hình chữ nhật với các cạnh (a_1, b_1) , (a_2, b_2) , (a_3, b_3) , (a_4, b_4) . Bạn hãy lập chương trình kiểm tra xem 4 hình này có thể ghép lại thành một hình vuông hay không.

2. Cho số thực φ ($0 < \varphi < \pi$). Một tia sáng xuất phát từ điểm $(1, 1)$ tạo với đường thẳng $x = 1$ một góc φ . Hãy xác định điểm tới của tia sáng trên trục Oy. Nếu góc $\varphi < 45^\circ$ thì tia sáng sẽ phản xạ qua trục Ox trước khi tới Oy.

3. Cho 3 số thực dương, hãy viết chương trình kiểm tra xem ba số này có tạo thành 3 cạnh của một tam giác, nếu có, thì tam giác là vuông, tù hay nhọn?

4. Cho hai điểm $A(x_1, y_1)$, $B(x_2, y_2)$ với $x_1 \neq x_2$. Hãy tìm trên trục hoành một điểm C sao cho tổng khoảng cách $CA+CB$ là nhỏ nhất.

5. Cho trước tọa độ 3 đỉnh của một hình chữ nhật là (x_1, y_1) , (x_2, y_2) , (x_3, y_3) . Hãy tìm tọa độ của đỉnh còn lại.

Trong các bài tập từ 6 đến 22, đầu vào là text file với dòng đầu tiên là số n, n dòng tiếp theo, mỗi dòng gồm một số số phân cách nhau bởi dấu cách, là các dữ liệu tương ứng với một chỉ số i. Kết quả thể hiện trên màn hình.

6. Cho trước số n và n đường thẳng: $ax + by + c_i = 0$, $i = 1, 2, \dots, n$. Hãy kiểm tra xem các khẳng định sau có đúng hay không:

- Có 2 đường thẳng song song.
- Có 2 đường thẳng trùng nhau.
- Có 3 đường thẳng đồng qui.

7. Cho trước điểm $A(x, y)$. Hãy tính tổng các khoảng cách từ điểm A tới các cạnh của hình chữ nhật với các đỉnh cho trước sau đây:

a. $(-0.5, -0.5)$, $(-0.5, 0.5)$, $(0.5, 0.5)$, $(0.5, -0.5)$.

b. $(0, 0)$, $(0, 1)$, $(1, 1)$, $(1, 0)$.

8. Trên trục số cho trước $n = 25$ khoảng $(a_1, b_1), (a_2, b_2), (a_3, b_3), \dots, (a_{25}, b_{25})$. Hỏi:

- Trên trục số có điểm nào thuộc tất cả các khoảng trên hay không? Nếu có thì hãy chỉ ra một điểm.
- Hợp của tất cả các khoảng trên có phải là một khoảng hay không? Nếu phải thì hãy chỉ ra tọa độ hai đầu mút của nó.
- Trên trục số có điểm nào thuộc ít nhất 3 trong số các khoảng trên hay không? Nếu có thì hãy chỉ ra một điểm.

9. Cho trước số n và n hình tròn $C_1, C_2, C_3, \dots, C_n$. Hình tròn C_i có tâm (x_i, y_i) và bán kính r_i cho trước. Hỏi trên mặt phẳng toạ độ có điểm nào thuộc tất cả các hình tròn trên hay không? Nếu có thì hãy chỉ ra một điểm.

10. Cho trước số n và n điểm $(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_n, y_n)$. Hãy kiểm tra xem khẳng định sau có đúng hay không: Với mỗi điểm (x_i, y_i) đều tồn tại một điểm (x_j, y_j) sao cho $n-2$ điểm còn lại đều nằm về 1 phía của đường thẳng đi qua (x_i, y_i) và (x_j, y_j) .

11. Trên mặt phẳng cho trước n điểm: $B_1, B_2, B_3, \dots, B_n$ (tọa độ cho trước). Một tam giác với các đỉnh trong $B_1, B_2, B_3, \dots, B_n$ được gọi là cực tiểu nếu nó không chứa một điểm nào khác của hệ này. Bạn hãy lập chương trình tìm ra một tam giác cực tiểu.

12. Cho trước số n chẵn và tập n điểm $(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_n, y_n)$, trong đó không có 2 điểm nào trùng nhau và không có 3 điểm nào thẳng hàng. Trung tuyến của tập là đường thẳng đi qua 2 điểm của tập và số các điểm nằm về 2 phía của đường thẳng đó là bằng nhau. Bạn hãy lập chương trình đếm số trung tuyến của tập.

13. Cho 3 đường tròn đồng qui tại O , 3 tia xuất phát từ O đi qua tâm A_1, A_2, A_3 của các hình tròn tạo với nhau những góc 120° . Giả sử các khoảng cách OA_1, OA_2, OA_3 đã biết trước. Hãy tính diện tích hình là hợp của 3 hình tròn.

14. Cho trước số n và n đường tròn đồng qui tại O , n tia xuất phát từ O đi qua tâm $A_1, A_2, \dots, A_n$ của các hình tròn lần lượt tạo với nhau những góc $\frac{2\pi}{n}$. Giả sử các khoảng cách $OA_1, OA_2, \dots, OA_n$ đã biết trước. Hãy tính diện tích hình là hợp của n hình tròn.

15. Trên mặt phẳng cho n hình chữ nhật có các cạnh song song với các trục tọa độ và cho bởi các cặp tọa độ trái trên và phải dưới: $((x_1, y_1), (z_1, t_1)), ((x_2, y_2), (z_2, t_2)), \dots, ((x_n, y_n), (z_n, t_n))$.

Tính diện tích phần mặt phẳng bị phủ bởi các hình chữ nhật nói trên.

16. Cho trước số n và n hình vuông tâm $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, cạnh $a_1, a_2, \dots, a_n$ tương ứng. Hãy tính diện tích hình là hợp của n hình vuông đó

17. Cho trước số n và n điểm $P_1(x_1, y_1), P_2(x_2, y_2), P_3(x_3, y_3), \dots, P_n(x_n, y_n)$, trong đó không có 2 điểm nào trùng nhau. Hãy tìm một đa giác lồi với đỉnh là các điểm trong số $P_1, P_2, P_3, \dots, P_n$ sao cho nó chứa tất cả các điểm còn lại.

18. Trên mặt phẳng cho n hình tròn với tọa độ tâm và bán kính là $((x_1, y_1), R_1), ((x_2, y_2), R_2), \dots, ((x_n, y_n), R_n)$.

Tính diện tích phần mặt phẳng bị phủ bởi các hình tròn nói trên.

19. Cho trước số n và n điểm $P_1(x_1, y_1), P_2(x_2, y_2), P_3(x_3, y_3), \dots, P_n(x_n, y_n)$. Hãy kiểm tra xem các khẳng định sau có đúng hay không:

- Đường gấp khúc $P_1P_2P_3\dots P_n$ không có điểm tự cắt.
- Đa giác $P_1P_2P_3\dots P_n$ là một đa giác lồi.

20. Cho trước 2 số thực x, y và tam giác có 3 đỉnh là $(0, 0), (0, 1)$ và (x, y) . Hãy tìm trong tam giác một điểm sao cho tổng các khoảng cách từ đó tới 3 đỉnh của tam giác là nhỏ nhất. Nếu các góc của tam giác không vượt quá 120° thì điểm cần tìm là điểm Toriceli, tức là điểm nhìn cả 3 cạnh của tam giác dưới một góc 120° , còn nếu tam giác có góc lớn hơn 120° thì điểm cần tìm chính là đỉnh với góc đó.

20. Cho trước số n và n điểm $P_1(x_1, y_1), P_2(x_2, y_2), P_3(x_3, y_3), \dots, P_n(x_n, y_n)$, trong đó không có 2 điểm nào trùng nhau. Hãy tìm một đa giác lồi với đỉnh là các điểm trong số $P_1, P_2, P_3, \dots, P_n$ sao cho nó chứa tất cả các điểm còn lại.

21. Cho hình chữ nhật CN_0 và n hình chữ nhật $CN_1, CN_2, \dots, CN_n$ với các cạnh song song với các trục tọa độ. Mỗi hình chữ nhật sẽ cho bởi tọa độ mặt phẳng của hai đỉnh đối diện.

Hãy viết chương trình kiểm tra xem tập các hình chữ nhật $CN_1, CN_2, \dots, CN_n$ có tạo thành một phủ rời nhau của CN_0 không?

22. Cho n hình chữ nhật $CN_1, CN_2, \dots, CN_n$ với các cạnh song song với các trục tọa độ.

Mỗi hình chữ nhật sẽ cho bởi tọa độ mặt phẳng của hai đỉnh đối diện.

- Tính diện tích phần mặt phẳng phủ bởi n hình chữ nhật trên.
- Hãy chỉ ra một điểm nằm trên mặt phẳng nằm trong nhiều hình chữ nhật nhất. Chỉ ra số các hình chữ nhật đó.

Bài 11. Ôn tập

Bài tập

1. Cho hai file f và g (không rõ kiểu). Hãy so sánh nội dung hai file đó.
2. Vẽ một hình vuông đặc quay quanh tâm của mình với vận tốc đều trong khi tâm hình vuông chuyển động tròn trên màn hình với vận tốc đều.
3. Vẽ một hình vuông đặc quay quanh tâm của mình với vận tốc đều trong khi tâm hình vuông chuyển động theo hình elip trên màn hình.

4. Lập chương trình vẽ đường xoắn ốc:

$$x = r \cos t$$

$$y = r \sin t$$

với $r = \frac{t}{2}$, $0 < t < 2k\pi$.

5. Lập chương trình vẽ hình bông hoa:

$$x = (A-B)\cos t - D\cos u$$

$$y = (A-B)\sin t - D\sin u$$

với: $D < B < A$ là 3 số tự nhiên cho trước,

$$n = \frac{B}{\text{UCLN}(A,B)}, \quad 0 < t < 2n\pi, \quad u = \frac{At}{B}.$$

6. Cho trước (x_1, y_1) , (x_2, y_2) là tọa độ của 2 điểm A và B. Hãy dựng hình vuông ABCD.

7. Vẽ trên màn hình một đoạn thẳng tự quay quanh điểm giữa cố định của mình.

8. Hãy hiển thị một số điểm chuyển động xiên trên màn hình với vận tốc khác nhau.

9. Hãy hiển thị một số điểm chuyển động xiên trên màn hình với vận tốc khác nhau, khi gặp cạnh màn hình sẽ phản xạ lại.

10. Danh sách cán bộ của cơ quan chứa trong file bản ghi f, dữ liệu về mỗi cán bộ được chứa trong một bản ghi gồm 4 trường mang thông tin về:

- Họ đệm: một xâu 25 kí tự.

- Tên: một chuỗi 10 kí tự.
- Số điện thoại: một số nguyên 7 chữ số.
- Địa chỉ: một chuỗi 40 kí tự.

Hãy lập chương trình tìm số điện thoại và địa chỉ của những cán bộ có tên cho trước

11. Cho trước n số thực dương $r_1, r_2, \dots, r_n$ là bán kính của n hình tròn tâm ở góc tọa độ và m điểm $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$. Hãy vẽ các hình đã cho và đếm số điểm nằm trong mỗi hình tròn.

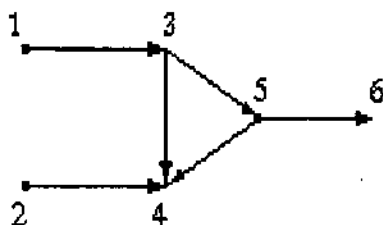
12. Cho một file bản ghi danh sách và một file chỉ mục tương ứng. Hãy kiểm tra trong danh sách xem có một người có họ tên cho trước hay không. Yêu cầu phải dùng thuật toán tìm nhị phân với sự trợ giúp của file chỉ mục (xem P20615, P20616).

13. Cho hai file: f là file kiểm kê chứa thông tin về hàng tồn kho gồm: Tên hàng, Số lượng tồn kho; g là file hoá đơn gồm: Tên hàng, Số lượng nhập, Số lượng xuất. Hãy dựa vào file g để cập nhật file f .

14. Trong một cuộc giao lưu trong một nhóm 7 bạn, mỗi bạn có trao đổi số điện thoại với một số bạn khác, cụ thể:

- A trao đổi số điện thoại với B và C
 - B trao đổi số điện thoại với A và C
 - C trao đổi số điện thoại với A, B và F
 - D trao đổi số điện thoại với E và G
 - E trao đổi số điện thoại với D và F
 - F trao đổi số điện thoại với C, E, và G
 - G trao đổi số điện thoại với D và F.
- a. Bạn hãy vẽ đồ thị mô tả quan hệ đó.
 - b. Bạn hãy lập ma trận kề của đồ thị đã vẽ.
 - c. Bạn hãy lập danh sách kề của đồ thị đã vẽ.
 - d. Bạn B muốn nhắn một tin cho bạn G thì phải nhắn qua bạn nào? Bạn hãy dùng chương trình lập ở bài P20415 để tìm một cách nhắn tin từ B đến G.

15. Từ một đồ thị có hướng gồm n đỉnh có thể lập hai ma trận nhị phân cấp n : Ma trận kề và ma trận liên thông. Phần tử a_{ij} của ma trận kề $A = (a_{ij})$ bằng 1 khi và chỉ khi có một cung đi từ đỉnh i tới đỉnh j . Phần tử b_{ij} của ma trận liên thông $B = (b_{ij})$ bằng 1 khi và chỉ khi có một đường đi đi từ đỉnh i tới đỉnh j . Ví dụ với đồ thị sau:

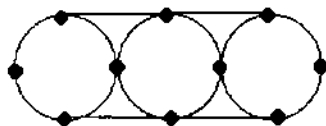
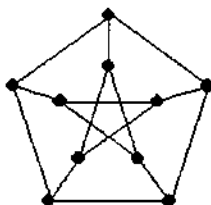
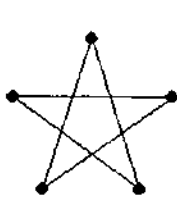


thì ma trận kề và ma trận liên thông sẽ là:

0	0	1	0	0	0	0	0	1	1	1	1
0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	1	1	0	0	0	0	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	1	0	1
0	0	0	0	0	0	0	0	0	0	0	0

Cho trước ma trận kề. Hãy lập chương trình tạo ma trận liên thông. Dữ liệu nhập vào từ tệp IN.DAT và kết quả đưa ra tệp OUT.DAT có cùng khuôn dạng.

16. Hãy mô tả ma trận kề của các đồ thị sau:



Bài 12. Bài toán duyệt. Thuật toán quay lui

Bài tập

1. Để liệt kê tất cả các dãy nhị phân độ dài n ta có thể làm theo cách sau:

Bước 1. Tạo mảng b độ dài n như một ngăn xếp để chứa 1 dãy nhị phân. Đặt $i := 1$ (xét phần tử đầu tiên).

Bước 2. Lần lượt cho $b[i]$ các giá trị từ 0 tới 1 (dùng vòng For). Với mỗi giá trị của $b[i]$:

- Nếu $i < n$ thì tăng i lên 1 đơn vị (xét phần tử tiếp theo) và quay lại bước 2.
- Nếu $i = n$ thì in dãy b .

Ví dụ với $n = 3$ thì ta phải thực hiện lần lượt như sau:

Thứ tự	i	b	Chú thích
1	1	0##	
2	2	00#	
3	3	000	In kết quả
4	3	001	In kết quả
5	2	01#	Tăng j ở vòng For thứ 2
6	3	010	In kết quả
7	3	011	In kết quả
8	1	1##	Tăng j ở vòng For thứ 1
9	2	10#	
10	3	100	In kết quả
11	3	101	In kết quả
12	2	11#	Tăng j ở vòng For thứ 2
13	3	110	In kết quả
14	3	111	In kết quả

Chương trình liệt kê tất cả các dãy nhị phân độ dài n theo cách trên có thể viết bằng kỹ thuật đệ quy như sau:

```
Program P21201; {quay lui: liệt kê dãy nhị phân}
Uses Crt;
Var i,n: Integer;
    b: array[1..100] of 0..1;
Procedure Inketqua;
```

```

Var i: integer;
Begin
For i:=1 to n do write(b[i]:3);
Writeln;
End;
Procedure Thu(i: integer);
Var j: integer;
Begin
For j:=0 to 1 do
begin
    b[i]:=j;
    if i=n then Inketqua else Thu(i+1);
end;
End;
BEGIN
Write('Nhap n: '); Readln(n);
Thu(1);
Readln;
END.

```

Bạn hãy chạy thử chương trình.

Trong các bài tập từ 2 đến 13, các số m và n được nhập từ bàn phím, kết quả đưa ra text file, mỗi dòng một tổ hợp (tập hợp).

2. Cho 5 chữ số khác nhau: 1, 2, 3, 4, 5 và số tự nhiên $n < 7$. Bạn hãy lập chương trình sử dụng kỹ thuật đệ qui như bài trên liệt kê tất cả các số khác nhau có n chữ số lập nên từ các chữ số đã cho.

3. Cho 5 chữ số khác nhau: 0, 1, 2, 3, 4 và số tự nhiên n. Bạn hãy lập chương trình liệt kê tất cả các số khác nhau có n chữ số lập nên từ các chữ số đã cho.

4. Bảng chữ cái có 9 chữ cái khác nhau: a, b, c, d, e, f, g, h, i. Mỗi "từ" gồm 3 chữ cái, trong đó có ít nhất 1 nguyên âm. Bạn hãy lập chương trình liệt kê và đếm tất cả các "từ" khác nhau lập nên từ bảng chữ cái đã cho.

5. Bảng chữ cái có 9 chữ cái khác nhau: a, b, c, d, e, f, g, h, i. Mỗi "từ" gồm từ 2 đến 4 chữ cái, trong đó có ít nhất 1 nguyên âm. Bạn hãy lập

chương trình liệt kê và đếm tất cả các "từ" khác nhau lập nên từ bảng chữ cái đã cho.

6. Để liệt kê tất cả các hoán vị của n số $1, 2, \dots, n$ ta có thể làm theo cách sau:

Bước 1. Tạo mảng b độ dài n như một ngăn xếp để chứa 1 hoán vị. Đặt $i:=1$ (xét phần tử đầu tiên).

Bước 2. Lần lượt cho $b[i]$ các giá trị từ 1 tới n (dùng vòng For) mà không trùng với các giá trị đã dùng từ $b[1]$ đến $b[i-1]$. Với mỗi giá trị của $b[i]$:

- Nếu $i < n$ thì tăng i lên 1 đơn vị (xét phần tử tiếp theo) và quay lại bước 2.
- Nếu $i = n$ thì in dãy b .

Ví dụ với $n = 3$ thì ta phải thực hiện lần lượt như sau:

Thứ tự	i	b	Chú thích
1	1	1##	
2	2	12#	
3	3	123	In kết quả
4	2	13#	Tăng j ở vòng For thứ 2
5	3	132	In kết quả
6	1	2##	Tăng j ở vòng For thứ 1
7	2	21#	
8	3	213	In kết quả
9	2	23#	
10	3	231	In kết quả
11	1	3##	Tăng j ở vòng For thứ 1
12	2	31#	
13	3	312	In kết quả
14	2	32#	Tăng j ở vòng For thứ 2
15	3	321	In kết quả

Chương trình liệt kê tất cả các hoán vị của n số $1, 2, \dots, n$ theo cách trên có thể viết bằng kỹ thuật đệ qui như sau:

```

Program P21208; {quay lui: liệt kê hoán vị}
Uses Crt;
Var  n: byte;

```


b: array[1..100] of byte;

Procedure Result;

Var i: byte;

Begin

For i:=1 to n do write(b[i]:3);

Writeln;

End;

Function Trung(i,j:byte): boolean;

Var k: byte;

Begin

trung:=false;

For k:=1 to i-1 do if b[k]=j then trung:=true;

End;

Procedure Try(i: integer);

Var j: byte;

Begin

For j:=1 to n do

if not trung(i,j) then

begin

b[i]:=j;

if i=n then Result else Try(i+1);

end;

End;

BEGIN

Write('Nhap n: '); Readln(n);

Try(1);

Readln;

END.

Bạn hãy chạy thử chương trình và so sánh với bài 1.

7. Bạn hãy cải tiến chương trình bài 6 để liệt kê tất cả các hoán vị của các tổ hợp m số từ n số 1, 2, ..., n. Các số n, m nhập từ bàn phím.

8. Bạn hãy cài tiến chương trình trên để liệt kê tất cả các tổ hợp m số từ n số $1, 2, \dots, n$. Các số n, m nhập từ bàn phím.

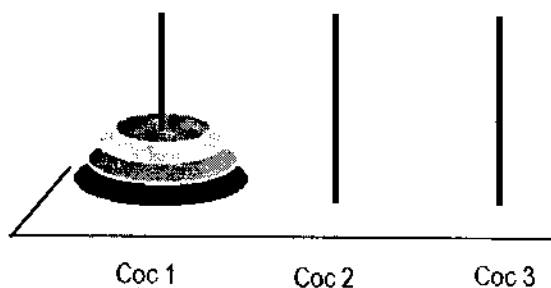
9. Cho trước một tập hợp gồm n số nguyên nhập từ bàn phím. Bạn hãy lập chương trình liệt kê tất cả các tập hợp con của tập đã cho.

10. Cho trước một dãy gồm n số nguyên nhập từ bàn phím. Bạn hãy lập chương trình liệt kê tất cả các dãy con của dãy đã cho.

11. Bạn hãy dùng chương trình đã lập ở bài trên để tìm dãy con đơn điệu tăng dài nhất của dãy đã cho.

12. Bài toán "Tháp Hà Nội":

Cho 3 cọc và n đĩa (lồng vào cọc) đặt ở cọc số 1 và theo thứ tự to dưới nhỏ trên (xem hình vẽ).



Hãy chuyển n đĩa đó sang cọc số 3 sao cho thứ tự vẫn được bảo toàn. Được sử dụng cọc số 2 làm trung gian và thỏa mãn yêu cầu khi chuyển thì chỉ cho phép đĩa bé đặt trên đĩa to. Mỗi lần được chuyển một đĩa. Hãy lập chương trình mô tả quá trình chuyển đổi đó. Số n nhập từ bàn phím.

13. Trong bài toán "Tháp Hà Nội". gọi H_n là số lần phải chuyển đĩa để thực hiện được yêu cầu bài toán. Hãy chứng minh công thức đệ qui sau: $H_n = 2H_{n-1} + 1$ với mọi $n > 1$.

Từ đó suy ra $H_n = 2^n - 1$.

14. Bài toán "8 hậu": Hãy sắp xếp 8 con hậu trên bàn cờ vua 8×8 sao cho không con hậu nào khống chế con nào. Hãy liệt kê tất cả các cách xếp như vậy.

Bài toán "8 hậu" sử dụng thuật toán quay lui để duyệt toàn bộ các phương án xếp hậu chấp nhận được. Tại đây chúng tôi mô tả 3 cách cài đặt của bài toán này.

Cài đặt 1:

Lời giải của bài toán là mảng $col[i]$ với $i=1..8$ chỉ ra vị trí hàng của con hậu nằm trên cột i .

Mô phỏng thuật toán:

```
i = 1
col[i] = 1
While i >= 1 do
  While col[i] <= 8 do col[i] = col[i] + 1
    {kiểm tra điều kiện col[i] của bài toán}
    For j := 1 to i-1 do
      If vị trí hậu (i,col[i]) bị hậu (j,col[j]) kiểm soát thì
        Tạm thời không thành công.
        Kiểm tra cột tiếp theo
      EndIf
    EndFor
    {Nếu tạm thời thành công}
    If i < 8
      col[i+1] = 0
      i = i + 1
    Else
      In kết quả col[i]
    EndIf
  EndWhile
  {quay lui}
  i = i - 1
EndWhile
```

Cài đặt 2:

Lời giải của bài toán là mảng $col[i]$ với $i=1..8$ chỉ ra vị trí hàng của con hậu nằm trên cột i . Mảng $Lib[k,j]$ mô tả các giá trị cho phép thứ j của con hậu trên cột k . Giá trị $Lib[k,j]=0$ có nghĩa là j là vị trí cho phép của hậu trên cột k . Trường hợp $Lib[k,j]=i > 0$ có nghĩa j là vị trí không cho phép của hậu trên cột k và i là chỉ số của con hậu đầu tiên cản trở điều đó.

Mô phỏng thuật toán:

```

{Ban đầu tất cả các ô đều tự do}
For u:=1 to 8 do For v:=1 to 8 do Lib[u,v]=0
i=1
col[i]=1
While i >=1 do
While col[i] <= 8 do
Giả sử col[i] - phần tử đầu tiên sao cho lib[i,col[i]] = 0
Gán col[i] = giá trị 0 đầu tiên từ lib[i,j], ngược lại gán
col[i]=9
j=col[i]
lib[i,j] = i
For k:=i+1 to 8 do
For l:=1 to 8 do
If lib[k,l] = 0 thì
If hậu tại (k,l) bị đe dọa bởi (i,j) thì đặt lib[k,l] = i
EndIf
EndIf
EndFor
EndFor
If i <8 thì i=i+1
Else In kết quả col[i]
EndIf
EndWhile
{Xác định lại các ô tự do}
For k:=i+1 to 8 do
For l:=1 to 8 do
If Lib[k,l] = i then Lib[k,l] =0
EndIf
EndFor
EndFor
{Quay lui}
Lib[i,col[i]]=i-1
i=i-1
EndWhile

```

Cài đặt 3:

Mô phỏng thuật toán:

n=0 {Số các vị trí đã chọn}

```

m=0 {Số hậu đã xếp}
For i:=1 to 8 do
  For j:=1 to 8 do
    col[i]=0
    Lib[i,j]=0
  EndFor
EndFor
i=1
While i >= 1 {Chọn hàng mới}
  Giả sử ii - chỉ số của hàng chứa ít 0 nhất
  nz(ii) số các 0 trên hàng ii
  If nz(ii) >= 1
  If nz(ii) > 1 then i=i+1 EndIf
  ch[i] = ii
  EndIf
  m=m+1 {đặt thêm một hậu nữa}
  col[ii] = số 0 đầu tiên của các ô tự do Lib[ii,j]
  Lib[ii,col[ii]] = 1
  For k:=i+1 to 8 do
    If col[k]=0 then
      For l:=1 to 8 do
        If Lib[k,l]=0 then
          If ô (k,l) bị hậu (i,j) uy hiếp then
            Lib[k,l]=i
          EndIf
        EndIf
      EndFor
    EndFor
  EndIf
  EndIf
  EndFor
  If m=8 then
    In kết quả col[i]
    Kết thúc chương trình
  Else
    {Quay lui}
    For k:=i+1 to 8 do
      If Col[k]=0 then
        For l:=1 to 8 do

```

```

If Lib[k,l]=i then Lib[k,l]=0 EndIf
EndFor
EndIf
EndFor
Lib[i,col[i]]=i-1
i=i-1
EndIf
EndWhile

```

Hãy viết các chương trình mô tả các cách cài đặt trên và so sánh tốc độ của các cài đặt này.

15. Bài toán "xếp ba lô": Có n vật với khối lượng và giá trị đã biết. Hãy chọn các vật để xếp vào ba lô với điều kiện tổng khối lượng được xếp không vượt quá một số cho trước, còn tổng giá trị đạt được là lớn nhất.

Hãy lập chương trình để giải bài toán trên.

Dữ liệu được cho trong một text file với qui cách như sau:

- Dòng đầu tiên là số n và số chỉ khối lượng giới hạn của ba lô.
- n dòng tiếp theo, mỗi dòng có 2 số tương ứng là khối lượng và giá trị của vật.

Kết quả đưa ra màn hình.

16. Hãy lập chương trình để xếp 8 con tượng lên bàn cờ tướng sao cho mỗi ô trên bàn cờ bị đúng một con tượng kiểm soát. Kết quả thể hiện trên màn hình.

17. Ma trận Latinh. Ma trận vuông $N \times N$ được gọi là ma trận Latinh nếu mỗi hàng và mỗi cột của ma trận đều lập thành một hoán vị của $1, 2, \dots, N$.

- a. Hãy viết chương trình tạo ra ma trận Latinh bậc 5.
- b. Hãy viết chương trình in ra tất cả các ma trận Latinh bậc 5.

18. Hãy viết chương trình xếp 5 con hậu trên bàn cờ vua 8×8 sao cho các con hậu này không chế toàn bộ các ô của bàn cờ. Kết quả thể hiện trên màn hình.

19. Mê cung được cho bởi mảng $A = (a_{ij})$ kích thước $M \times N$. Biết $a_{ij} = 0$ nếu vị trí (i,j) là tự do, và $a_{ij} = 1$ nếu vị trí (i,j) là đóng kín. Từ vị trí ban đầu là một điểm tự do có thể dịch chuyển sang điểm tự do ở bên cạnh. Nếu ra được điểm bên ngoài thì ta nói là có thể thoát khỏi mê cung. Bạn hãy lập chương trình kiểm tra xem từ một điểm tự do cho trước có thể

thoát ra khỏi mê cung hay không. Trong trường hợp có thể hãy chỉ ra con đường thoát khỏi mê cung đó. Dữ liệu nhập từ tệp INPUT.TXT có dạng: dòng đầu tiên ghi hai số m, n , m dòng tiếp theo mỗi dòng n số 0 hoặc 1 là dữ liệu của mê cung. Vị trí ban đầu nhập từ bàn phím. Kết quả đặt trong file Output.out.

20. Bài toán "Người Gác rừng".

Một người Gác rừng có nhiệm vụ kiểm tra N khu vực (được đánh số từ 1 đến N). Thời gian đi từ địa điểm i đến j là C_{ij} (chú ý có thể $C_{ij} \neq C_{ji}$). Hàng ngày, người Gác rừng cần xuất phát từ một địa điểm (chẳng hạn 1), đi qua tất cả các khu vực và quay trở về vị trí xuất phát.

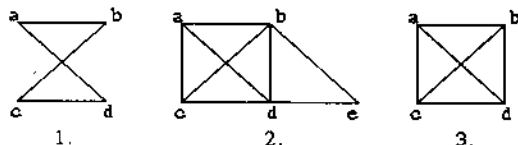
Bài toán yêu cầu chỉ ra một cách đi của người Gác rừng sao cho tổng thời gian đi là nhỏ nhất.

Mã trận dữ liệu ban đầu (C_{ij}) được đưa vào từ tệp INPUT.DAT. Kết quả thể hiện trên màn hình.

Bài 13. Đường đi Euler

Câu hỏi và bài tập

1. Đường đi Euler của một đồ thị là đường đi qua mỗi cạnh của đồ thị đúng một lần. Bạn hãy giải thích vì sao trong các đồ thị sau:



các đường đi: a, b, c, d, a trong đồ thị 1; a, b, c, d, e, b, d, a, c trong đồ thị 2, là các đường đi Euler, còn các đường đi: a, b, c, d trong đồ thị 1; b, a, c, d, e, b, d, a, c, b trong đồ thị 2; a, b, c, d, a, c trong đồ thị 3 không phải là các đường đi Euler.

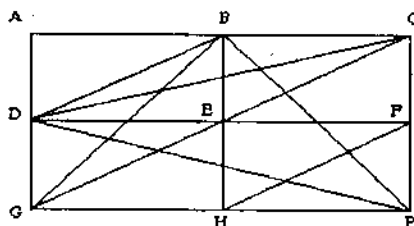
2. Một đường đi Euler khép kín (có đỉnh đầu trùng với đỉnh cuối) gọi là một chu trình Euler. Ví dụ như đường đi a, b, c, d, a trong đồ thị ở bài 1.

Định lý Euler

Một đồ thị vô hướng liên thông có chu trình Euler khi và chỉ khi mọi đỉnh của nó đều có bậc chẵn, tức là tổng số cạnh xuất phát từ mỗi đỉnh là số chẵn.

Hãy kiểm tra xem trong 3 đồ thị ở hình trên, đồ thị nào có chu trình Euler.

3. Một bác đưa thư cho một khu phố có sơ đồ như sau:



Hàng ngày, xuất phát từ nhà bưu điện (P), bác phải đi qua mỗi phố ít nhất một lần. Bạn hãy tìm cho bác một con đường đi tiết kiệm thời gian nhất.

4. Chứng minh định lý EULER được phát biểu trong bài 2.

5. Một đồ thị không có chu trình Euler vẫn có thể có đường đi Euler như đồ thị 2 bài 1. Bạn hãy phát biểu một tiêu chuẩn tương tự định lý Euler để nhận dạng một đồ thị có đường đi Euler.

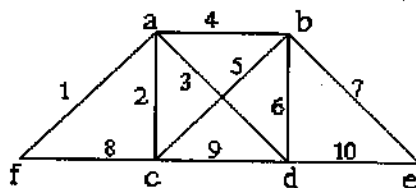
6. Để tìm một chu trình Euler, ta có thể làm như sau:

- Tạo một mảng CE để ghi đường đi và một ngăn xếp Stack để xếp các đỉnh ta sẽ xét. Xếp vào đó một đỉnh tùy ý u của đồ thị, tức là ta sẽ xét u đầu tiên.
- Xét đỉnh trên cùng của ngăn xếp, giả sử đó là đỉnh v:
 - Nếu v là đỉnh cô lập thì lấy v khỏi ngăn xếp, đưa vào CE.
 - Nếu v liên thông với đỉnh x thì ta xếp x vào ngăn xếp, sau đó xoá bỏ cạnh (v,x).
- Quay lại bước trên cho tới khi ngăn xếp rỗng thì dừng.

Kết quả đường đi được chứa trong CE theo thứ tự ngược lại.

Ví dụ

Với đồ thị:

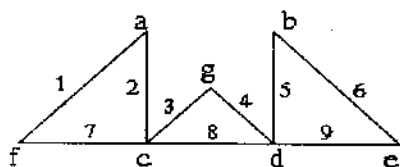


ta sẽ thực hiện các bước với kết quả như sau:

Bước	Stack	CE	Cạnh còn lại
1	f		1,2,3,4,5,6,7,8,9,10
2	f,a		2,3,4,5,6,7,8,9,10
3	f,a,c		3,4,5,6,7,8,9,10
4	f,a,c,f		3,4,5,6,7,9,10
5	f,a,c	f	3,4,5,6,7,9,10
6	f,a,c,b	f	3,4,6,7,9,10

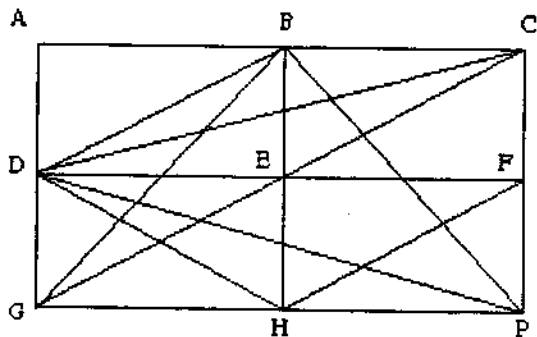
7	f,a,c,b,d	f	3,4,7,9,10
8	f,a,c,b,d,c	f	3,4,7,10
9	f,a,c,b,d	f,c	3,4,7,10
10	f,a,c,b,d,e	f,c	3,4,7
11	f,a,c,b,d,e,b	f,c	3,4
12	f,a,c,b,d,e,b,a	f,c	3
13	f,a,c,b,d,e,b,a,d	f,c	
14	f,a,c,b,d,e,b,a	f,c,d	
15	f,a,c,b,d,e,b	f,c,d,a	
16	f,a,c,b,d,e	f,c,d,a,b	
17	f,a,c,b,d	f,c,d,a,b,e	
18	f,a,c,b	f,c,d,a,b,e,d	
19	f,a,c	f,c,d,a,b,e,d,b	
20	f,a	f,c,d,a,b,e,d,b ,c	
21	f	f,c,d,a,b,e,d,b ,c,a	
22		f,c,d,a,b,e,d,b ,c,a,f	

Kết quả cuối cùng ta được chu trình Euler f,a,c,b,d,e,b,a,d,c,f.
Theo cách tương tự như trên, hãy tìm một chu trình Euler cho đồ thị sau:



7. Lập chương trình thể hiện thuật toán tìm chu trình Euler cho ở bài trên. Đầu vào của chương trình là một text file chứa ma trận kề của đồ thị. Chương trình phải kiểm tra xem đồ thị có chu trình Euler hay không; nếu có thì mới tìm một chu trình.

8. Bạn hãy tìm một đường đi Euler cho đồ thị sau:



9. Một đồ thị không có chu trình Euler vẫn có thể có đường đi Euler. Khi đó đồ thị có đúng 2 đỉnh có bậc lẻ, tức là tổng số cạnh xuất phát từ mỗi đỉnh đó là số lẻ. Một đường đi Euler phải xuất phát từ một trong 2 đỉnh đó và kết thúc ở đỉnh kia. Bạn hãy lập chương trình thể hiện thuật toán tìm đường đi Euler giống thuật toán cho ở hai bài trên. Đầu vào của chương trình là một text file chứa ma trận kề của đồ thị. Chương trình phải kiểm tra xem đồ thị có đường đi Euler hay không; nếu có thì mới tìm một đường đi Euler.

10. Hãy chứng minh Định lý EULER tổng quát:

Đồ thị liên thông G có đường đi EULER khi và chỉ khi G có nhiều nhất là 2 đỉnh có bậc lẻ.

11. Để liệt kê tất cả các đường đi Euler của một đồ thị n đỉnh, m cạnh, ta có thể dùng kỹ thuật đệ qui như sau:

Bước 1. Tạo mảng b độ dài $m+1$ như một ngăn xếp để chứa 1 đường đi. Đặt $b[0] := 1$ và $i := 1$ (xét đỉnh thứ 1 của đường đi).

Bước 2. Lần lượt cho $b[i]$ các giá trị là các đỉnh kề với $b[i-1]$ mà cạnh $(b[i-1], b[i])$ không trùng với các cạnh đã dùng từ $b[0]$ đến $b[i-1]$. Với mỗi giá trị của $b[i]$:

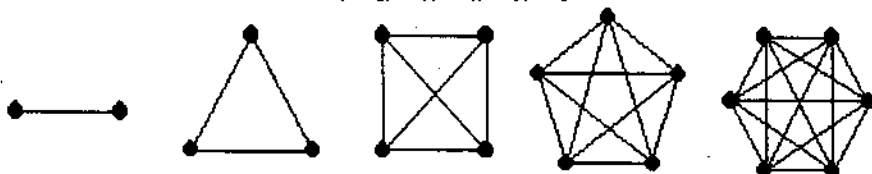
- Nếu $i < m$ thì tăng i lên 1 đơn vị (xét đỉnh tiếp theo) và quay lại bước 2.
- Nếu $i = m$ thì in dãy b .

Bạn hãy lập chương trình liệt kê tất cả các đường đi Euler của một đồ thị.

12. Đồ thị K_n

Đồ thị K_n là đồ thị đầy đủ bao gồm n đỉnh, trong đó hai đỉnh bất kỳ đều được nối với nhau.

Hình vẽ sau mô tả các đồ thị K_2, K_3, K_4, K_5, K_6 :



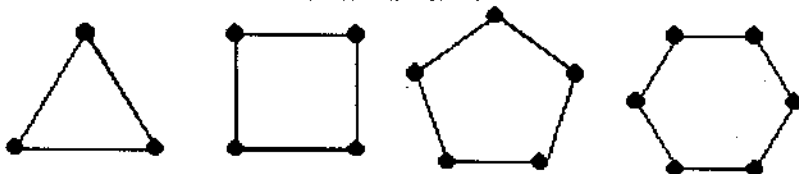
- Tìm điều kiện để K_n có đường đi EULER.
- Tìm điều kiện để K_n có chu trình EULER.
- Trong các trường hợp có thể, hãy chỉ ra một chu trình hoặc đường đi EULER của đồ thị này.

13. Đồ thị C_n

C_n là đồ thị bao gồm n đỉnh $v_1, v_2, \dots, v_n$ với n cạnh sau:

$$(v_1, v_2), (v_2, v_3), \dots, (v_{n-1}, v_n), (v_n, v_1).$$

Hình vẽ sau mô tả các đồ thị C_3, C_4, C_5, C_6 :

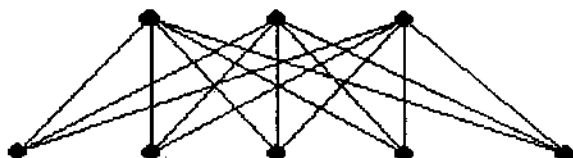
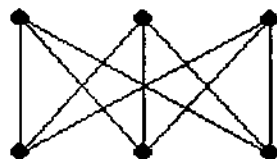
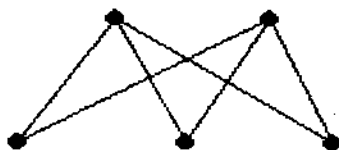


- Tìm điều kiện để C_n có đường đi EULER.
- Tìm điều kiện để C_n có chu trình EULER.
- Trong các trường hợp có thể, hãy chỉ ra một chu trình hoặc đường đi EULER của đồ thị này.

14. Đồ thị $K_{n,m}$

Đồ thị $K_{n,m}$ bao gồm 2 nhóm đỉnh, nhóm thứ nhất có n đỉnh, nhóm thứ hai có m đỉnh. Mọi đỉnh của nhóm thứ nhất đều kết nối với mọi đỉnh của nhóm thứ hai, ngoài ra không có bất cứ loại kết nối nào khác.

Hình vẽ sau mô tả các đồ thị $K_{2,3}, K_{3,3}$ và $K_{3,5}$:

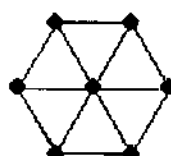
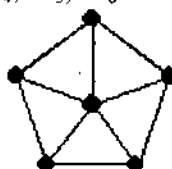
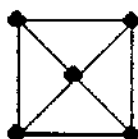


- Tìm điều kiện để $K_{n,m}$ có đường đi EULER.
- Tìm điều kiện để $K_{n,m}$ có chu trình EULER.
- Trong các trường hợp có thể, hãy chỉ ra một chu trình hoặc đường đi EULER của đồ thị này.

15. Đồ thị W_n

Đồ thị W_n bao gồm C_n cộng thêm một đỉnh là "tâm", đỉnh này có kết nối với tất cả các đỉnh còn lại.

Hình vẽ sau mô tả các đồ thị W_3, W_4, W_5, W_6 :



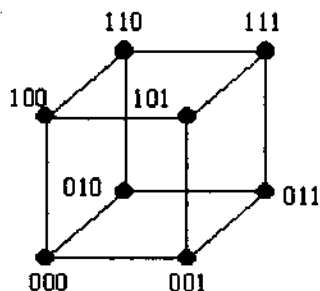
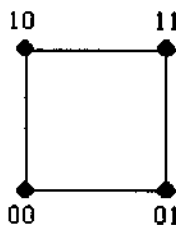
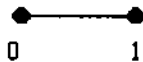
- Tìm điều kiện để W_n có đường đi EULER.
- Tìm điều kiện để W_n có chu trình EULER.
- Trong các trường hợp có thể, hãy chỉ ra một chu trình hoặc đường đi EULER của đồ thị này.

16. Đồ thị Q_n (n- Cube):

Đồ thị Q_n là đồ thị có các đỉnh mô tả 2^n xâu nhị phân (bit) độ dài n .

Hai đỉnh là kề (và có kết nối) khi và chỉ khi biểu diễn nhị phân của chúng chỉ khác nhau đúng một bit.

Hình vẽ sau mô tả các đồ thị Q_1, Q_2 và Q_3 :

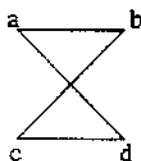


- Tìm điều kiện để Q_n có đường đi EULER.
- Tìm điều kiện để Q_n có chu trình EULER.
- Trong các trường hợp có thể, hãy chỉ ra một chu trình hoặc đường đi EULER của đồ thị này.

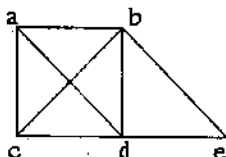
Bài 14. Chu trình Hamilton

Câu hỏi và bài tập

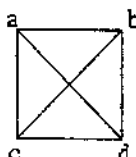
1. Đường đi Hamilton của một đồ thị là một đường đi qua mỗi đỉnh của đồ thị đúng một lần. Bạn hãy giải thích vì sao trong các đồ thị sau:



1.



2.



3.

thì các đường đi: a, b, c, d trong đồ thị 1; a, b, c, d, e trong đồ thị 2, là đường đi Hamilton, còn các đường đi: a, b, c trong đồ thị 1; b, a, c, d, e, b, d, a, c, b trong đồ thị 2; a, b, c, d, a, c trong đồ thị 3 không phải là đường đi Hamilton.

2. Một chu trình (tức đường đi khép kín) bắt đầu từ một đỉnh nào đó, qua mỗi đỉnh của đồ thị đúng một lần rồi quay trở về đỉnh ban đầu thì được gọi là một *chu trình Hamilton*. Bạn hãy kiểm tra xem trong 3 đồ thị ở hình trên, đồ thị nào có chu trình Hamilton.

3. Để liệt kê tất cả các chu trình Hamilton của một đồ thị n đỉnh, ta có thể làm như sau:

Bước 1. Tạo mảng b độ dài $n+1$ như một ngăn xếp để chứa 1 đường đi. Đặt $b[0] := 1$ và $i := 1$ (xét đỉnh thứ 1 của chu trình).

Bước 2. Lần lượt cho $b[i]$ các giá trị từ 1 tới n (dùng vòng For) là các đỉnh kề với $b[i-1]$ mà không trùng với các đỉnh đã đi qua từ $b[0]$ đến $b[i-1]$. Với mỗi giá trị của $b[i]$:

- Nếu $i < n$ thì tăng i lên 1 đơn vị (xét đỉnh tiếp theo) và quay lại bước 2.
- Nếu $i = n$ và $b[i] = b[0]$ thì in dãy b .

Hãy lập chương trình liệt kê tất cả các chu trình Hamilton của một đồ thị.

4. Hãy chứng minh định lý Hamilton sau đây:

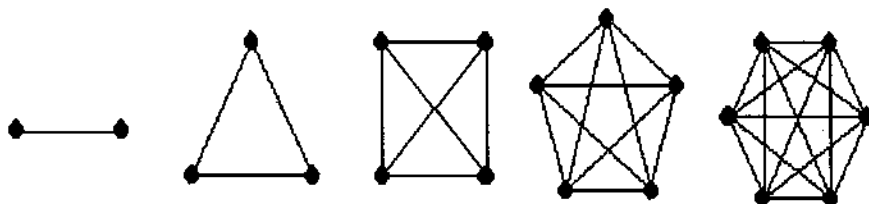
Đồ thị liên thông G với số đỉnh $n > 2$ có chu trình Hamilton khi mọi đỉnh của nó đều có bậc không nhỏ hơn $n/2$.

5. Bạn hãy lập chương trình liệt kê tất cả các đường đi Hamilton của một đồ thị.

6. Đồ thị K_n

Đồ thị K_n là đồ thị đầy đủ bao gồm n đỉnh, trong đó hai đỉnh bất kỳ đều được nối với nhau.

Hình vẽ sau mô tả các đồ thị K_2, K_3, K_4, K_5, K_6 :



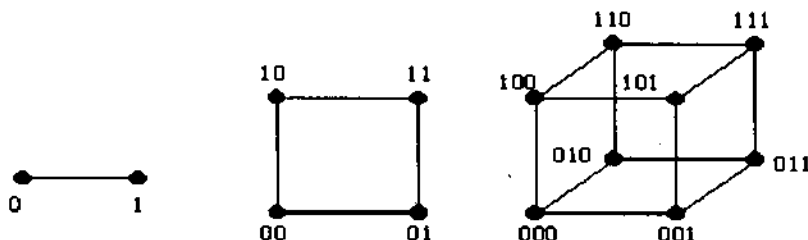
Chứng minh rằng đồ thị K_n có chu trình Hamilton với mọi $n > 2$.

7. Đồ thị Q_n (n - Cube)

Đồ thị Q_n là đồ thị có các đỉnh mô tả 2^n xâu nhị phân (bit) độ dài n .

Hai đỉnh là kề (và có kết nối) khi và chỉ khi biểu diễn nhị phân của chúng chỉ khác nhau đúng một bit.

Hình vẽ sau mô tả các đồ thị Q_1, Q_2 và Q_3 :

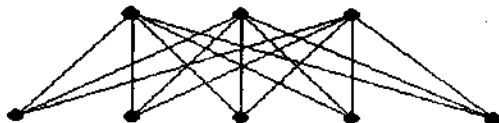
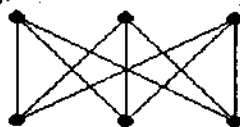
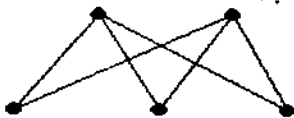


Chứng minh rằng đồ thị Q_n luôn có chu trình Hamilton với mọi $n > 1$.

8. Đồ thị $K_{n,m}$

Đồ thị $K_{n,m}$ bao gồm 2 nhóm đỉnh, nhóm thứ nhất có n đỉnh, nhóm thứ hai có m đỉnh. Mọi đỉnh của nhóm thứ nhất đều kết nối với mọi đỉnh của nhóm thứ hai, ngoài ra không có bất cứ loại kết nối nào khác.

Hình vẽ sau mô tả các đồ thị $K_{2,3}$, $K_{3,3}$ và $K_{3,5}$:

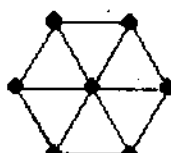
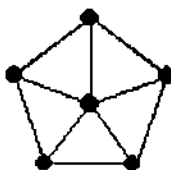
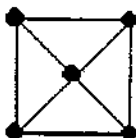
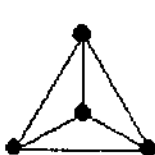


- Tìm điều kiện để $K_{n,m}$ có đường đi Hamilton.
- Tìm điều kiện để $K_{n,m}$ có chu trình Hamilton.
- Trong các trường hợp có thể, hãy chỉ ra một chu trình hoặc đường đi Hamilton của đồ thị này.

9. Đồ thị W_n

Đồ thị W_n bao gồm C_n cộng thêm một đỉnh là "tâm", đỉnh này có kết nối với tất cả các đỉnh còn lại.

Hình vẽ sau mô tả các đồ thị W_3 , W_4 , W_5 , W_6 :



- Tìm điều kiện để W_n có đường đi Hamilton.
- Tìm điều kiện để W_n có chu trình Hamilton.
- Trong các trường hợp có thể, hãy chỉ ra một chu trình hoặc đường đi Hamilton của đồ thị này.

10. Một đồ thị G được gọi là *kép* nếu tập các đỉnh V của đồ thị có thể tách thành hai tập con rời nhau $V = V_1 + V_2$ sao cho mọi cạnh của đồ thị đều kết nối một đỉnh nào đó của V_1 với một đỉnh của V_2 , hay nói cách khác không có cạnh nào của đồ thị nằm hoàn toàn trong V_1 hoặc V_2 . Ví dụ đồ thị $K_{n,m}$ là một đồ thị kép. Chứng minh rằng một đồ thị kép với số đỉnh lẻ sẽ không có chu trình Hamilton.

11. Xét chuyển động của con mã trên bàn cờ vua kích thước $n \times m$. Với đỉnh là các ô của bàn cờ, chuyển động hợp lý của con mã là cạnh, ta sẽ

tạo ra một đồ thị hoàn chỉnh với n đỉnh. Hãy mô phỏng đồ thị của chuyển động của con mã trên bàn cờ 3×3 .

12. Tương tự bài trên, hãy mô phỏng đồ thị của chuyển động con mã trên bàn cờ 3×4 .

13. Chứng minh rằng đồ thị mô phỏng chuyển động của con mã trên bàn cờ $n \times m$ là đồ thị kép.

14. Với n, m là lẻ, chứng minh rằng trên bàn cờ $n \times m$ không tồn tại một đường đi khép kín của con mã sao cho mỗi ô đi qua đúng một lần.

Trong các bài sau, dữ liệu vào được nhập từ bàn phím, kết quả thể hiện trên màn hình.

15. Bạn hãy lập chương trình tìm một đường đi khép kín của con mã trên bàn cờ 8×8 sao cho mỗi ô được đi qua đúng 1 lần.

16. Bạn hãy lập chương trình liệt kê tất cả các đường đi của một con mã trên bàn cờ 8×8 sao cho mỗi ô được đi qua đúng 1 lần.

17. Trên một lưới $N \times N$ ô vuông, Kanguru có thể bước như sau: $(x, y) \rightarrow (x+1, y-1)$ hoặc $(x, y) \rightarrow (x-5, y+7)$ nhưng không được ra khỏi lưới. Hãy xác định xem từ những điểm ban đầu nào Kanguru có thể đi đến được tất cả các ô khác của lưới, mỗi ô chỉ được đi qua một lần.

18. Cho hình lập phương, mỗi mặt của nó được đánh số bằng một số tự nhiên nào đó (ký hiệu là $a_1, a_2, \dots, a_6$). Ta xét một đường đi khép kín, xuất phát từ một mặt nào đó, đi qua tất cả 6 mặt và trở về chỗ cũ.

Quy tắc đi là chỉ có quyền vượt qua 1 cạnh để sang mặt bên cạnh. Mỗi đường đi khép kín như vậy tương ứng với một thứ tự các số của mặt đã đánh dấu: $b_{1,6} b_{2,6} b_{3,6} b_{4,6} b_{5,6} b_{6,6} b_1$.

- Tìm một đường đi khép kín làm cực tiểu hóa hàm giá trị: $\sum_{i=1}^6 ib_i$

- Tìm một đường đi để cực đại hóa hàm giá trị trên.

19. Có thể xếp tất cả 28 quân domino thành một dãy được hay không? Nếu được, hãy chỉ ra một cách xếp.

20. Giả sử đã có một quân domino trên mặt bàn. Hãy lập chương trình xếp các quân domino tuần tự về hai phía sao cho số quân domino còn lại trên tay có số điểm bé nhất.

Bài 15. Các bài toán trên đồ thị

Câu hỏi và bài tập

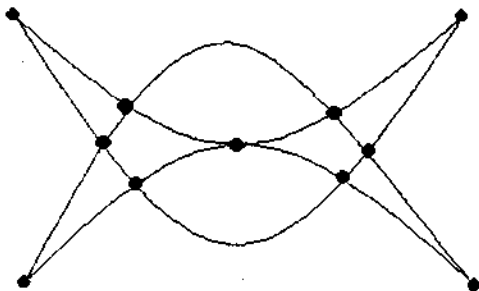
1. Đồ thị được gọi là "đơn liên" nếu có thể được vẽ lại bằng bút chì theo một nét, nghĩa là không nhấc bút và không có cạnh nào được vẽ bằng 2 nét. Khẳng định sau coi như đã biết: Đồ thị là "đơn liên" khi và chỉ khi số các đỉnh mà từ đó xuất phát số lẻ cạnh, không lớn hơn 2.

Cho trước ma trận kề của một đồ thị.

- Xác định xem đồ thị có là "đơn liên" hay không.
- Nếu đồ thị là "đơn liên" thì hãy chỉ ra một cách vẽ một nét cho đồ thị đó.

2. Hãy chứng minh định lý về đồ thị đơn liên được mô tả trong bài trên.

3. Hãy vẽ hình sau bằng một nét.



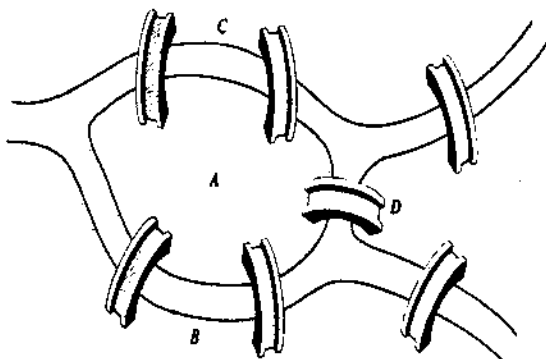
4. Bài toán thành phố cổ Konigberg nổi tiếng thế kỷ 18.

Thành phố cổ Konigberg được chia thành 4 khu vực A, B, C, D bởi 7 cái cầu như hình vẽ. Có hay không đường đi để đi qua tất cả 7 chiếc cầu trên, mỗi cầu chỉ được đi qua đúng một lần.

Bạn hãy giúp người thành phố cổ giải quyết bài toán trên..

5. Trong trò chơi Lines, người ta đặt trên bảng vuông 8×8 ô vuông một số hòn bi, mỗi hòn bi chiếm một ô vuông. Muốn chuyển một hòn bi từ ô s đến ô t phải giải 3 bài toán:

- Trong ô s có hòn bi nào hay không và ô t có còn trống không?



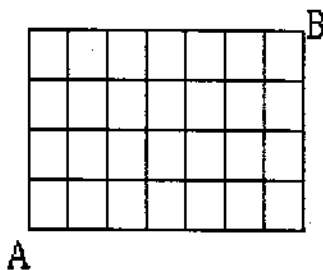
b. Có đường đi, tức là một dãy ô trống nằm kề nhau (chung cạnh), từ s đến t hay không?

c. Nếu có thì đường đi nào là ngắn nhất?

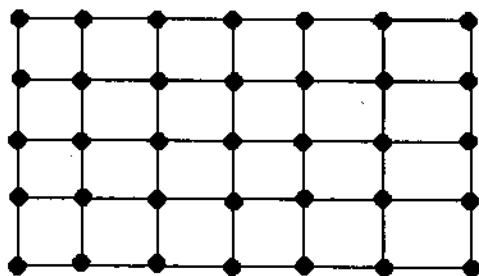
Biết trạng thái của bảng được chứa trong một mảng nhị phân A kích thước 8×8 cho trước, số 0 biểu thị một ô trống, số 1 biểu thị một hòn bi. Bạn hãy lập chương trình chuyển một hòn bi từ ô s đến ô t với s và t nhập từ bàn phím.

6. Cho đồ thị vô hướng G với ma trận kề A. Cho trước 2 đỉnh i, j của ma trận. Ta nói một đường đi từ i đến j có độ dài k nếu đường đi này phải đi qua đúng k cạnh khác nhau của G. Gọi $B = A^k$. Chứng minh rằng phần tử b_{ij} của ma trận B chính bằng số các đường đi khác nhau độ dài k từ i đến j của đồ thị G.

7. Cho một lưới ô vuông kích thước $m \times n$ (m, n nhập từ bàn phím). Lập chương trình tính số các đường đi ngắn nhất từ A đến B và chỉ đi trên các cạnh của lưới. Kết quả thể hiện trên màn hình.



8. Đồ thị sau có hay không một đường đi (chu trình) Euler, Hamilton?



9. Bài toán "Tìm đường đi": Có n thành phố được đánh số thứ tự $1, 2, \dots, n$. Giữa hai thành phố bất kì có thể có một đường đi trực tiếp (không qua thành phố khác) hoặc không có. Cho trước ma trận đối xứng $A = (a_{ij})$ chứa dữ liệu về các đường đi trực tiếp, trong đó a_{ij} là độ dài đường đi trực tiếp từ thành phố i đến thành phố j ($a_{ij} = 0$ nếu giữa hai thành phố không có đường đi trực tiếp). Hãy lập chương trình tìm độ dài của đường đi ngắn nhất từ thành phố 1 đến thành phố n . Dữ liệu nhập từ tệp văn bản INPUT.DAT.

10. Ma trận liên thông: Cho đồ thị G với ma trận kề A bậc n . Ma trận liên thông C là ma trận thỏa mãn tính chất sau:

$C = (C_{ij})$, ở đây $C_{ij} = 1$ khi và chỉ khi từ đỉnh i tồn tại đường đi đến j .

Thuật toán sau mô tả cách tính C (Thuật toán Warshall):

Ta xây dựng dãy các ma trận $B^{(k)} = (B^{(k)}_{ij})$ như sau:

$$B^{(0)} = A.$$

Với $k = 1, 2, \dots, n$, $B^{(k)}$ được xây dựng đệ quy theo công thức:

$$B^{(k)}_{ij} = B^{(k-1)}_{ij} \vee (B^{(k-1)}_{ik} \wedge B^{(k-1)}_{kj}).$$

Ký hiệu ở đây $\vee$ là phép hợp logic, $\wedge$ là phép nhân logic được định nghĩa theo bảng sau:

$1 \vee 1 = 1$	$1 \wedge 1 = 1$
$1 \vee 0 = 1$	$1 \wedge 0 = 0$
$0 \vee 1 = 1$	$0 \wedge 1 = 0$
$0 \vee 0 = 0$	$0 \wedge 0 = 0$

Khi đó ta sẽ có $C = B^{(n)}$.

Hãy chứng minh tính đúng đắn của thuật toán trên.

11. Cho lưới $m \times n$ ô vuông, trong mỗi ô cho trước một số tự nhiên. Hãy tìm cách chia lưới trên làm hai phần (chia theo cạnh lưới) sao cho trị tuyệt đối các hiệu số của tổng các số trong mỗi phần có giá trị nhỏ nhất.

				7	
	1	3	5		
	12	2	5		
	9	2	10		

Dữ liệu về lưới ô vuông nhập từ tệp LUOI.DAT. Các số $m, n \leq 20$.

12. Thuật toán Kruskal tìm đường đi ngắn nhất trên đồ thị.

Cho đồ thị G với n đỉnh. Ta định nghĩa ma trận A như sau (ma trận kề tổng quát):

Nếu giữa hai đỉnh i, j tồn tại đường nối thì A_{ij} = độ dài của đường nối đó, ngược lại ta đặt $A_{ij} = 0$.

Ma trận đường đi $C = (C_{ij})$ được định nghĩa như sau:

C_{ij} = khoảng cách của đường đi ngắn nhất giữa đỉnh i và j , trong trường hợp ngược lại thì $C_{ij} = 0$. Thuật toán Kruskal (hay còn gọi là thuật toán Warshall tổng quát) mô tả cách tạo ma trận C như sau:

Ta xây dựng dãy các ma trận $B^{(k)} = (B^{(k)}_{ij})$ như sau:

$$B^{(0)} = A.$$

Với $k = 1, 2, \dots, n$, $B^{(k)}$ được xây dựng đệ qui theo công thức

$$B^{(k)}_{ij} = \min(B^{(k-1)}_{ij}, B^{(k-1)}_{ik} + B^{(k-1)}_{kj}).$$

Khi đó $C = B^{(n)}$.

Hãy chứng minh tính đúng đắn của thuật toán trên.

13. Có n thành phố. Bạn hãy lập chương trình nối các thành phố bằng các con đường sao cho từ một thành phố bất kỳ có thể đi đến thành phố bất kỳ khác chỉ phải qua cùng lắm là 1 thành phố trung gian nữa, và số các đường là ít nhất.

14. Thuật toán Dijkstra tìm khoảng cách của đường đi ngắn nhất trên đồ thị vô hướng có trọng lượng.

Cho đồ thị G với các đỉnh là $a = v_0, v_1, v_2, \dots, v_n = b$ đỉnh. Ta định nghĩa ma trận A như sau (ma trận kề tổng quát):

Nếu giữa hai đỉnh i, j tồn tại đường nối thì a_{ij} = độ dài của đường nối đó. ngược lại ta đặt $a_{ij} = 0$.

Thuật toán sử dụng hàm $L(v_i)$ chỉ ra khoảng cách ngắn nhất từ đỉnh a đến đỉnh v_i của đồ thị.

$L(a) := 0$

For $i:=1$ to n do $L(v_i) := 0$

EndFor

$S := \emptyset$

While $b \notin S$ do

$u :=$ đỉnh không nằm trong S với $L(u)$ nhỏ nhất

$S := S + \{u\}$

For $v \in S$

If $L(u) + A_{uv} < L(v)$ then $L(v) := L(u) + A_{uv}$

EndFor

EndWhile

Kết quả $L(b)$ - khoảng cách ngắn nhất từ a đến b .

Hãy chứng minh tính đúng đắn của thuật toán trên.

15. Bài toán sên bò.

Trên lưới ô vuông xuất phát từ một đỉnh $(0,0)$ cần phải đi đến điểm kết thúc tại $(N,0)$ (N là số tự nhiên cho trước).

Qui tắc đi: Mỗi bước $(x_1, y_1) \rightarrow (x_2, y_2)$ thoả mãn điều kiện:

- $x_1 \leq x_2 \leq x_1 + 1$,
- $y_1 \leq y_2 \leq y_1 + 1$ (sên bò).
- Tìm một cách đi sao cho trong quá trình đi nó có thể lên cao nhất trên trục tung (tức là tọa độ y đạt cực đại).
- Lập chương trình thống kê tất cả các cách đi của sên từ vị trí ban đầu $(0,0)$ đến vị trí kết thúc $(N,0)$.

Hãy thể hiện lưới ô vuông trên màn hình. Kết quả của bài toán thể hiện ngay trên lưới này.

16. Thuật toán chỉ ra một đường đi ngắn nhất trên đồ thị có trọng lượng (thuật toán Dijkstra tổng quát).

Cho đồ thị G với các đỉnh là $a = v_0, v_1, v_2, \dots, v_n$ đỉnh. Ta định nghĩa ma trận A như sau (ma trận kề tổng quát): Nếu giữa hai đỉnh i, j tồn tại đường nối thì a_{ij} = độ dài của đường nối đó, ngược lại ta đặt $a_{ij} = 0$.

Thuật toán cần chỉ ra một đường đi từ a đến một đỉnh bất kỳ b nằm trên đồ thị.

Thuật toán sử dụng các hàm số $L(v)$ chỉ ra khoảng cách ngắn nhất từ a đến v , hàm logic $KQ(v)$ chỉ ra rằng đã tìm được đường đi ngắn nhất từ a đến v , hàm $Parent(v)$ chỉ ra đỉnh sẽ được nối với v trên đường đi ngắn nhất.

```
For i:=1 to n do L(v):=0
KQ(v):=false
EndFor
L(a):=0
KQ(a):=True
last:=a
While not KQ(b) do
For v ∈ G do
If not KQ(v) and L(v) > L(last) + Alast, v
then begin L(v):=L(last)+Alast, v
Parent(v):=last
end
EndIf
u:= vertex (đỉnh) with (với) KQ(u)=false and (và) L(u)
is minimum (nhỏ nhất)
KQ(u):=True
last:=u
EndWhile
```

Khi đó đường đi ngắn nhất từ a đến b sẽ là:

$a, Parent(a), Parent(Parent(a)), \dots, b$

Hãy chứng minh tính đúng đắn của thuật toán trên.

17. Cho bảng vuông $2n \times 2n$ ô, trên mỗi dòng và mỗi cột có đúng hai ô được đánh dấu. Nối các vị trí đánh dấu theo dòng và theo cột ta được các đường đi.

a. Chứng minh rằng các đường nối này sẽ tạo ra các đường khép kín (nghĩa là từ một ô được đánh dấu, đi theo đường nối ta sẽ quay trở về vị trí ban đầu).

b. Hãy chỉ ra một cách chọn các ô đánh dấu sao cho các đường đi tạo thành đúng một đường khép kín.

c. Viết chương trình tạo ra tất cả các phương án đánh dấu các ô sao cho chúng tạo ra một đường đi khép kín.

PHẦN II. CHƯƠNG TRÌNH

Bài 1. Mảng nhiều chiều

```
1. Program P20101a;
Uses Crt;
Const nmax=50;
      inp = 'INPUT.DAT';
Type
      Mang = array [1..nmax,1..nmax] of real;
Var a,b: Mang;
      n,i,j,k,l: integer;
Procedure Nhap;
Var i,j: integer; f: text;
Begin
  Assign(f,inp); Reset(f);
  Readln(f,n);
  For i:=1 to n do
    begin
      For j:=1 to n do Read(f,a[i,j]);
      Readln(f);
    end;
  Close(f);
End;

Procedure Xuat(a: Mang);
Var i,j: integer;
Begin
  For i:=1 to n do
    begin
      For j:=1 to n do Write(a[i,j]:8:2);
      Writeln;
    end;
End;
```

19.09.86
86.8.86

```

BEGIN
  Nhap;
  For i:=1 to n do
    For j:=1 to n do
      For k:=1 to n do
        For l:=1 to n do
          If (k>i)and(l>j) then b[i,j]:=b[i,j]+a[k,l];
        Clrscr;
      Xuat(a);
      Writeln;
    Xuat(b);
    Readln;
  END.

```

3. Program P20103;

{Ptu max nam o goc trai tren, ptu min o goc phai duoi}

Uses Crt;

Const nmax = 50;

inp = 'INPUT.DAT';

Type

Mang=array [1..nmax,1..nmax] of real;

Var a: Mang;

n,i,j,il,jl: integer;

Procedure Nhap;

Var i,j: integer; f: text;

Begin

Assign(f,inp); Reset(f);

Readln(f,n);

For i:=1 to n do

begin

For j:=1 to n do Read(f,a[i,j]);

Readln(f);

end;

Close(f);
End;

Procedure Xuat(a: Mang);
Var i,j: integer;
Begin
 For i:=1 to n do
 begin
 For j:=1 to n do Write(a[i,j]:8:2);
 Writeln;
 end;
 End;

Procedure Doihang(k,l: integer);
Var i: integer; tg: real;
Begin
 For i:=1 to n do
 begin
 tg:=a[k,i];
 a[k,i]:=a[l,i];
 a[l,i]:=tg;
 end;
 End;

Procedure Doicot(k,l: integer);
Var i: integer; tg: real;
Begin
 For i:=1 to n do
 begin
 tg:=a[i,k];
 a[i,k]:=a[i,l];
 a[i,l]:=tg;
 end;
 End;

```

BEGIN
  Nhap;
  ClrScr;
  Xuat(a);
  Writeln;
  (* Tim phan tu lon nhat *)
  i1:=1; j1:=1;
  For i:=1 to n do
    For j:=1 to n do
      If a[i1,j1]<a[i,j] then
        begin
          i1:=i;
          j1:=j
        end;
    Doihang(1,i1); Doicot(1,j1);
  (* Tim phan tu nho nhat *)
  i1:=1; j1:=1;
  For i:=1 to n do
    For j:=1 to n do
      If a[i1,j1]>a[i,j] then
        begin
          i1:=i;
          j1:=j
        end;
  If (i1=1) then
    begin
      Writeln('Khong thuc hien duoc vi max va min cung
hang');
      Readln; Halt;
    end;
  If (j1=1) then
    begin

```

```

        Writeln('Khong thuc hien duoc vi max va min cung
cot');
        Readln; Halt;
    end;
    Doihang(n,i1); Doicot(n,j1);
    Xuat(a);
    Readln;
END.

```

5. Program P20105; {lap mang vuong}

Uses Crt;

Var a: array [1..7,1..7] of integer;

 i,j,hang: integer;

BEGIN

 Clrscr;

 hang:=2;

 (* Vong ngoai *)

 For i:=1 to 6 do

 begin

 a[1,i]:=i;

 a[i,7]:=i+6;

 a[7,8-i]:=i+12;

 a[8-i,1]:=i+18;

 end;

 (* Vong thu hai *)

 For i:=2 to 5 do

 begin

 a[2,i]:=i+23;

 a[i,6]:=i+27;

 a[6,8-i]:=i+31;

 a[8-i,2]:=i+35;

 end;

 (* Vong thu ba *)

```

For i:=3 to 4 do
  begin
    a[3,i]:=i+38;
    a[i,5]:=i+40;
    a[5,8-i]:=i+42;
    a[8-i,3]:=i+44;
  end;
a[4,4]:=49;
ClrScr;
For i:= 1 to 7 do
  For j:=1 to 7 do
    begin
      Gotoxy((j-1)*6+10,hang+(i*2));
      Write(a[i,j]:5);
    end;
  Readln;
END.

```

7. Program P20107; {Lap mang B mot chieu tu mang A hai chieu}

```

Uses Crt;
Const  nmax = 8;
       inp = 'INPUT.DAT';
Type
  Mang1 = array [1..nmax*nmax] of real;
  Mang2 = array [1..nmax,1..nmax] of real;
Var  a: Mang2; b: Mang1;
     n,i,j,k,l: integer;

```

```

Procedure Nhap;
Var  i,j: integer;
     f: text;
Begin
  Assign(f,inp); Reset(f);

```

```

Readln(f,n);
For i:=1 to n do
  begin
    For j:=1 to n do Read(f,a[i,j]);
    Readln(f);
  end;
Close(f);
End;

```

```

Procedure Xuat1(a: Mang1);
Var i: integer;
Begin
  For i:=1 to Sqr(n) do Write(a[i]:8:2);
End;

```

```

Procedure Xuat2(a: Mang2);
Var i,j: integer;
Begin
  For i:=1 to n do
    begin
      For j:=1 to n do Write(a[i,j]:8:2);
      Writeln;
    end;
End;

```

```

BEGIN
  Nhap;
  ClrScr;
  Xuat2(a);
  Writeln;
  (* xet cac duong cheo phu *)
  l:=0;
  For k:=1 to 2*n-1 do
    If (k mod 2)=1 then

```

```

If  $k \leq n$  then
    begin
         $i := k; j := 1;$ 
        Repeat
             $l := l + 1;$ 
             $b[l] := a[i, j];$ 
             $i := i - 1;$ 
             $j := j + 1;$ 
        Until  $i < 1;$ 
    end
Else { $k \leq n$ }
    begin
         $i := n; j := 1 + k - n;$ 
        Repeat
             $l := l + 1;$ 
             $b[l] := a[i, j];$ 
             $i := i - 1;$ 
             $j := j + 1;$ 
        Until  $i < 1 + k - n;$ 
    end
Else {( $k \bmod 2$ ) = 1}
    If  $k \leq n$  then
        begin
             $i := 1; j := k;$ 
            Repeat
                 $l := l + 1;$ 
                 $b[l] := a[i, j];$ 
                 $i := i + 1;$ 
                 $j := j - 1;$ 
            Until  $j < 1;$ 
        end
    Else { $k \leq n$ }
        begin
             $i := 1 + k - n; j := n;$ 

```



```

        Repeat
            l:=l+1;
            b[l]:=a[i,j];
            i:=i+1;
            j:=j-1;
        Until j<1+k-n;
    end;
Xuat1(b);
Readln;
END.

9. Program P20109; {lap mang 1 chieu tu mang 2 chieu}
Uses Crt;
Const  nmax = 50;
        inp = 'INPUT.DAT';
Type
    Mang1=array [1..nmax*nmax] of real;
    Mang2=array [1..nmax,1..nmax] of real;
Var  a: Mang2;
    O: Mang1;
    n,i,j,k,l,r,t,b: integer;

Procedure Nhap;
Var i,j: integer; f: text;
Begin
    Assign(f,inp); Reset(f);
    Readln(f,n);
    For i:=1 to n do
        begin
            For j:=1 to n do Read(f,a[i,j]);
            Readln(f);
        end;
    Close(f);
End;

```

```

Procedure Xuat1(a: Mang1);
Var i: integer;
Begin
  For i:=1 to Sqr(n) do Write(a[i]:8:2);
End;

```

```

Procedure Xuat2(a: Mang2);
Var i,j: integer;
Begin
  For i:=1 to n do
    begin
      For j:=1 to n do Write(a[i,j]:8:2);
      Writeln;
    end;
End;

```

```

Procedure Xep;
Begin
  Repeat
    O[k]:=a[i,j]; Inc(k);
    If (i=t)and(j<r) then Inc(j)
    Else If (j=r) and (i<b) then Inc(i)
      Else If (i=b)and(j>l) then Dec(j)
        Else If (j=l)and(i>t) then Dec(i);
  Until (i=t) and (j=l);
  Inc(l); Inc(t);
  Dec(r); Dec(b);
  i:=t; j:=l;
  If k<=Sqr(n) then Xep;
End;

```

```

BEGIN
  Nhap;

```

```

i:=1; j:=1; k:=1; l:=1;
r:=n; t:=1; b:=n;
Xep;
ClrScr;
Xuat2(a);
Writeln;
Xuat1(O);
Readln;
END.

```

11. Program P20111; {Lap mang moi bang doi cho cac khoi}

```
Uses Crt;
```

```
Const nmax = 50;
```

```
inp = 'INPUT.DAT';
```

```
Type
```

```
Mang=array [1..nmax,1..nmax] of real;
```

```
Var a,b: Mang;
```

```
n,i,j: integer;
```

```
Procedure Nhap;
```

```
Var i,j: integer;
```

```
f: text;
```

```
Begin
```

```
Assign(f,inp); Reset(f);
```

```
Readln(f,n);
```

```
For i:=1 to 2*n do
```

```
begin
```

```
For j:=1 to 2*n do Read(f,a[i,j]);
```

```
Readln(f);
```

```
end;
```

```
Close(f);
```

```
End;
```

```
Procedure Xuat(a: Mang);
```

Var i,j: integer;
 Begin
 For i:=1 to 2*n do
 begin
 For j:=1 to 2*n do Write(a[i,j]:8:2);
 Writeln;
 end;
End;

BEGIN
 Nhap;
 For i:=1 to n do
 For j:=1 to n do
 begin
 b[i,j]:=a[i+n,j];
 b[i,j+n]:=a[i,j];
 b[i+n,j+n]:=a[i,j+n];
 b[i+n,j]:=a[i+n,j+n];
 end;
 ClrScr;
 Xuat(a);
 Writeln;
 Xuat(b);
 Readln;
END.

13. Program P20113b;
 Uses Crt;
 Var b: array [1..10,1..10] of real;
 n,i,j: integer;

Procedure Nhap;
 Begin
 Clrscr;

```

Write('Nhap so N: '); Readln(N);
End;

Procedure Xuat;
Var i,j: integer;
Begin
  For i:=1 to n do
    begin
      For j:=1 to n do Write(b[i,j]:8:5);
      Writeln;
    end;
End;

```

```

BEGIN
  Nhap;
  For i:=1 to n do
    begin
      b[i,1]:=1/i;
      For j:=2 to n do b[i,j]:=b[i,j-1]/j;
    end;
  Xuat;
  Readln;
END.

```

15. Program P20115; {tim diem thuoc tat ca cac hinh chu nhac}

```

Uses Crt;
Const inp = 'P20115.INP';
Var a: array [1..4,1..100] of real;
    x1,y1,x2,y2: real;
    n,i: integer;

```

```

Procedure Nhap;
Var i: integer; f: text;
Begin

```

```

Assign(f,inp); Reset(f);
Readln(f,n);
For i:=1 to n do Readln(f,a[1,i],a[2,i],a[3,i],a[4,i]);
Close(f);
End;

```

```

Function Giao(i: integer): boolean;
Begin
  If x1<a[1,i] then x1:=a[1,i];
  If x2>a[3,i] then x2:=a[3,i];
  If y1<a[2,i] then y1:=a[2,i];
  If y2>a[4,i] then y2:=a[4,i];
  If (x1<=x2)and(y1<=y2) then Giao:=true Else Giao:=false;
End;

```

```

BEGIN
  Nhap;
  Clrscr;
  For i:=1 to n do Writeln(a[1,i]:8:1,a[2,i]:8:1,a[3,i]:8:1,a[4,i]:8:1);
  Writeln;
  x1:=a[1,1]; y1:=a[2,1];
  x2:=a[3,1]; y2:=a[4,1];
  i:=1;
  While (i<=n)and Giao(i) do Inc(i);
  If i>n then Write('Co diem thuoc tat ca cac hinh CN:',
x1:8:1,y1:8:1,x2:8:1,y2:8:1)
  Else Write('Khong co diem nao thuoc tat ca cac hinh CN. ');
  Readln;
END.

```

Bài 2. Các thao tác trên xâu kí tự

1. Program P20201; {Tao mang danh sach}

Uses Crt;

Const max = 100;

Var i,n: byte;

ten,holot: array [1..max] of string[20];

lop: array [1..max] of string[5];

danhsach: array [1..max] of string[45];

Procedure Nhap;

Var i: integer; f: text;

Begin

Assign(f,'HODEM.TXT'); Reset(f);

n:=0;

While not Eof(f) do

begin

Inc(n);

Readln(f,holot[n]);

end;

Close(f);

Assign(f,'TEN.TXT'); Reset(f);

For i:=1 to n do Readln(f,ten[i]);

Close(f);

Assign(f,'LOP.TXT'); Reset(f);

For i:=1 to n do Readln(f,lop[i]);

Close(f);

End;

Procedure Xuat;

Var i: integer;

Begin

For i:=1 to n do Writeln(danhsach[i]);

End;

```
BEGIN
  Nhap;
  For i:=1 to n do danh sach[i]:=holot[i]+' '+ten[i]+' '+lop[i];
  ClrScr;
  Xuat;
  Readln;
END.
```

3. Program P20203; {Sua loi chinh ta}

```
Uses Crt;
Const  Dau1: set of char = ['.',',',';',':','!'];
       Dau2: set of char = ['.',',','!'];
Var    p,i: word;
       S: string;
       finp,fout: text;
```

```
BEGIN
  Assign(finp,'Anphabet.TXT'); Reset(finp);
  Assign(fout,'KQ.TXT'); Rewrite(fout);
  While not Eof(finp) do
    begin
      Readln(finp,S);
      Repeat
        p:=Pos(' ',S);
        If p<>0 then Delete(S,p,1);
      Until p=0;
      i:=1;
      Repeat
        If S[i] in Dau1 then
          begin
            If S[i-1]=' ' then
              begin Delete(S,i-1,1); Dec(i); end;
            If S[i+1]<>' ' then Insert(' ',S,i+1);
          end;
      Until
```



```

        If S[i] in Dau2 then S[i+2]:=Uppcase(S[i+2]);
        Inc(i);
    Until i=Length(S);
    Writeln(fout,S);
end;
Close(finp); Close(fout);
END.

```

5. Program P20205; {Giong hang phai}

Uses Crt;

Const inp='P20204.INP';

out='P20204.OUT';

Var Max,m,n,i,j,l,a,b: word;

S: string;

finp,fout: text;

Procedure RTrim(Var S: string);

Begin

While S[Length(S)] = #32 do Delete(S,Length(S),1);

End;

Function Rong: Word;

Var Max: integer;

Begin

Assign(finp,inp); Reset(finp);

Max:=0;

While not Eof(finp) do

begin

Readln(finp,S);

RTrim(S);

n:=0; m:=0; i:=1; l:=Length(S);

While (S[i] = #32) and (i <= l) do Inc(i);

Repeat

If S[i] = #32 then

24
08
1996

```

        begin
            While (S[i]= #32)and(i<=l) do begin Inc(n);
Inc(i); end;
            Inc(m);
        end;
        While (S[i]<>#32)and(i<=l) do Inc(i);
        Until i>l;
        If Max<l-n+m then Max:=l-n+m;
    end;
    Close(finp);
    Rong:=Max;
End;

```

```

BEGIN
    ClrScr;
    Max:=Rong; {Do rong van ban}
    Assign(finp,inp); Reset(finp);
    Assign(fout,out); Rewrite(fout);
    While not SeekEof(finp) do {Xu li tung dong}
        begin
            Readln(finp,S);
            RTrim(S); {Tinh so lieu cua dong}
            n:=0; m:=0; i:=1; l:=Length(S);
            While (S[i]= #32)and(i<=l) do Inc(i);
            Repeat
                If S[i]= #32 then
                    begin
                        While (S[i]= #32)and(i<=l) do begin Inc(n);
Inc(i); end;
                        Inc(m);
                    end;
                While (S[i]<>#32)and(i<=l) do Inc(i);
            Until i>l;
            l:=l-n;
        end;
    end;

```

```

a:=(Max-1) div m; b:=(Max-1) mod m;
{Xu li dong}
i:=1; m:=0;
While (S[i]= #32)and(i<=l) do Inc(i);
Repeat
  If S[i]= #32 then
    begin
      Inc(m);
      If m<=b then n:=a+1 Else n:=a;
      While (s[i]= #32)and(i<=Length(S)) do
Delete(S,i,1);
      For j:=1 to n do Insert(' ',S,i);
      i:=i+n;
    end;
  While (s[i]<>#32)and(i<=Length(S)) do Inc(i);
Until i>=Max;
Writeln(fout,S);
end;
Close(finp); Close(fout);
END.

```

7. Program P20207;

{Ma hoa xau bang cach tang thu tu tu ma hoa trong bang chu cai len 1 }

Uses Crt;

Const khoa=1;

Var s1,s2: string;

i: byte;

BEGIN

ClrScr;

Write('Nhap xau ki tu can ma hoa: ');Readln(s1);

Write('Nhap xau ki tu can giai ma: ');Readln(s2);

Clrscr;

Writeln('Ma hoa cua "',s1,'" la :');

For i:=1 to length(s1) do
 If s1[i] in ['A'..'Z'] then s1[i]:=chr((((ord(s1[i])-39)+khoa) mod 26)+65)
 Else If s1[i] in ['a'..'z'] then s1[i]:=chr((((ord(s1[i])-71)+khoa) mod 26)+97);
 Writeln(s1);
 Writeln('Giai ma cua "',s2,'" la :');
 For i:=1 to length(s2) do
 If s2[i] in ['A'..'Z'] then s2[i]:=chr((((ord(s2[i])-39)-khoa) mod 26)+65)
 Else If s2[i] in ['a'..'z'] then s2[i]:=chr((((ord(s2[i])-71)-khoa) mod 26)+97);
 Writeln(s2);
 Readln;
 END.

9. Program P20209; {ma hoa theo vong tron}

Uses Crt;

Var s1,s2: string;

i,khoa: byte;

BEGIN

Clrscr;

Write('Khoa ma la bao nhieu : '); Readln(khoa);

Write('Nhap xau ki tu can ma hoa : '); Readln(s1);

Write('Nhap xau ki tu can giai ma : '); Readln(s2);

Clrscr;

Writeln('Ma hoa cua "',s1,'" la :');

For i:=1 to length(s1) do

If s1[i] in ['A'..'Z'] then s1[i]:=chr((((ord(s1[i])-39)+khoa) mod 26)+65)

Else If s1[i] in ['a'..'z'] then s1[i]:=chr((((ord(s1[i])-71)+khoa) mod 26)+97);

Writeln(s1);

Writeln('Giai ma cua "',s2,'" la :');

```

For i:=1 to length(s2) do
  If s2[i] in ['A'..'Z'] then s2[i]:=chr((((ord(s2[i])-39)-khoa) mod
26)+65)
  Else If s2[i] in ['a'..'z'] then s2[i]:=chr((((ord(s2[i])-71)-khoa)
mod 26)+97);
  Writeln(s2);
  Readln;
END.

```

11. Program P20211; {Ma hoa xau ki tu}

Uses Crt;

```

Var i,j,l,d,c: word;
    s1,s2: string;
    a: array [1..20,1..20] of char;

```

Procedure Gan1(ch: char);

Begin

 If (c<>l) then Inc(c)

 Else begin c:=1; Inc(d); end;

 a[d,c]:=ch;

End;

Procedure Gan2(ch: char);

Begin

 If (d<>l) then Inc(d)

 Else begin d:=1; Inc(c); end;

 a[d,c]:=ch;

End;

BEGIN

 Clrscr;

 Write('Nhap xau can ma hoa: '); Readln(s1);

 Write('Nhap xau can giai ma: '); Readln(s2);

 Clrscr;

 l:=trunc(sqrt(length(s1)));

```

If l*l<length(s1) then Inc(l);
d:=1; c:=0;
For i:=1 to length(s1) do Gan1(s1[i]);
For i:=1 to l*l-length(s1) do Gan1(' ');
Writeln('Xau ban dau: ',s1);
Write('Xau ma hoa: ');
For i:=1 to l do
  For j:=1 to l do Write(a[j,i]);
Writeln;
l:=trunc(sqrt(length(s2)));
If l*l<length(s2) then Inc(l);
d:=0; c:=1;
For i:=1 to length(s2) do Gan2(s2[i]);
For i:=1 to l*l-length(s2) do Gan2(' ');
Writeln('Xau ban dau: ',s2);
Write('Xau ma hoa: ');
For i:=1 to l do
  For j:=1 to l do Write(a[i,j]);
Readln;
END.

```

13. Program P20213; {Kiem tra khoa va ma hoa van ban}

Uses Crt;

```

Var i,j,max,l:word;
    code: array [1..2,1..9] of byte;
    dl: array[ 1..300] of char;
    s: array [1..2] of string;
    khoa,tam,out: string;

```

Procedure Error;

Begin

Write('Khoa khong hop le');

Readln;

halt;

End;

```

Procedure Mahoa(xau: string; khoa: byte);
Begin
  For i:=1 to length(xau) do dl[i]:=xau[i];
  l:=length(xau);
  For i:=1 to max-(length(xau) mod max) do dl[l+i]:=' ';
  Inc(l,max-(length(xau) mod max));
  For i:=1 to l div max do
    begin
      tam:="";
      For j:=1 to max do tam:=tam+dl[(i-1)*max+j];
      For j:=1 to max do out[j]:=tam[code[khoa,j]];
      Write(out);
    end;
End;

```

```

BEGIN
  Clrscr;
  Write('Nhap khoa: '); Readln(khoa);
  max:=0; tam:="";
  If length(khoa)=0 then Error;
  For i:=1 to Length(khoa) do
    begin
      If pos(khoa[i], '123456789')=0 then Error;
      If pos(khoa[i], tam) < 0 then Error;
      tam:=tam+khoa[i];
      If max < ord(khoa[i]) then max:=ord(khoa[i]);
    end;
  max:=max-48;
  If max < length(khoa) then Error;
  For i:=1 to max do
    begin
      code[1,i]:=ord(khoa[i])-48;
      code[2,ord(khoa[i])-48]:=i;
    end;

```

668.6t

```

    end;
    Out:="";
    For i:=1 to max do out:=out+' ';
    Write('Nhap xau can ma hoa: '); Readln(s[1]);
    Write('Nhap xau can giai ma: '); Readln(s[2]);
    Clrscr;
    Writeln('Xau ban dau: ',s[1]);
    Write('Xau ma hoa: '); Mahoa(s[1],1); Writeln;
    Writeln('Xau ban dau: ',s[2]);
    Write('Xau giai ma: '); Mahoa(s[2],2);
    Readln;
END.

```

15. Program P20215; {Sua loi cho xau ki tu}

Uses Crt;

Var i: byte;

s: string;

BEGIN

Clrscr;

Write('Nhap xau ki tu: ');

Readln(s);

While pos('#',s) < 0 do

 If pos('#',s)=1 then Delete(s,pos('#',s),1)

 Else Delete(s,pos('#',s)-1,2);

Write('Xau da sua: ',s);

Readln;

END.

Bài 3. Kiểu tập hợp

1. a, f: đúng.

b, c, d, e, g, h: sai vì kiểu của phần tử không được phép (được phép: byte, char, ...).

3. c và e.

5. Program P20305; {ham dem so phan tu cua tap hop}

Uses Crt;

Type

 Tap = set of 0..99;

Const inp = 'Numbers.dat';

Var S: Tap;

 i: byte;

Procedure Nhap;

Var a: byte;

 f: text;

Begin

 S:=[];

 Assign(f, inp); Reset(f);

 While not SeekEof(f) do

 begin

 Readln(f, a);

 If (a>=0) and (a<=99) then S:=S+[a];

 end;

 Close(f);

End;

Function Card(S: Tap): byte;

Var i, n: byte;

Begin

 n:=0;

 For i:=0 to 99 do If i in S then Inc(n);

 Card:=n;

End;

BEGIN

Nhap;

Clrscr;

Write('Tap S co ',Card(S),' phan tu.');

Readln;

END.

7. Program P20307; {Liet ke cac phan tu cua tap hop}

Uses Crt;

Const inp = 'Numbers.dat';

Var S: set of byte;

i: byte;

Procedure Nhap;

Var a: byte; f: text;

Begin

S:=[];

Assign(f,inp); Reset(f);

While not f^{eof}

begin Readln(f,a); S:=S+[a]; end;

Close(f);

End;

BEGIN

Nhap;

Clrscr;

For i:=1 to 255 do If i in S then Write(i:8);

Readln;

END.

9. Program P20509; {Tap hop cac so nguyen chan khong chia het cho 3}

Uses Crt;

Var so: set of byte;

```

    i: byte;
BEGIN
    Clrscr;
    so:=[];
    For i:=0 to 255 do If i mod 2=0 then so:=so+[i];
    For i:=0 to 255 do If (i mod 3=0) then so:=so-[i];
    For i:=0 to 255 do If i in so then Write(i:4);
    Readln;
END.

```

11. Program P203011; {Tim cac so nguyen to kieu byte}

```

Uses Crt;
Var  sang: set of byte;
     i,j: byte;
BEGIN
    Clrscr;
    sang:=[];
    For i:=2 to 255 do sang:=sang+[i];
    For i:=2 to 255 do
        If i in sang then
            begin
                j:=2;
                While (i*j<=255) do
                    begin
                        sang:=sang-[i*j];
                        Inc(j);
                    end;
            end;
    Writeln('Cac so nguyen to:');
    For i:=2 to 255 do If i in sang then Write(i:4);
    Readln;
END.

```

13. Program P20313; {Nhap mot xau toan cac chu cai}

```

Uses Crt;

```

7A
588.6

```

Const A:set of char = ['a'..'z','A'..'Z'];
Var ch: char;
    st: string;
BEGIN
  Clrscr;
  st:="";
  Writeln('Nhap vao mot xau toan cac chu cai:');
  Repeat
    ch:=Readkey;
    If ch in A then begin st:=st+ch; write(ch); end
    Else if ch<>#13 then Write(#7);
    If ch = #0 then ch:=Readkey;
  Until ch = #13;
END.

```

15. Program P20315; {Nhap mot hoan vi}

```

Uses Crt;
Var n,i: byte;
    so: set of byte;
    Ok: boolean;
    a: array[1..126] of byte;
BEGIN
  Clrscr;
  Repeat
    Write('Nhap so n (0<n<126): '); Readln(n);
    If (n<1)or(n>125) then Writeln(#7,'Sai');
  Until (n>=1)and(n<=125);
  so:=[];
  Writeln('Nhap mot hoan vi cua cac so tu 1 den ',n,' ');
  For i:=1 to n do
    begin
      Repeat
        Write('Nhap phan tu thu ',i,' '); Readln(a[i]);
        If (a[i] in so)or(n<a[i])or(a[i]<1) then Ok:=false Else
          Ok:=true;
      Until Ok;
    end
  end

```

```

    Until Ok;
    so:=so+[a[i]];
end;
For i:=1 to n do Write(a[i]:8); Readln;
END.

```

17. Program P20317; {loai bo cac phan tu khoi tap hop}

Uses Crt;

Var tap: set of char;

ch: char;

BEGIN

Clrscr;

tap:=[];

Writeln('Nhap cac phan tu cho mot tap hop cac ki tu: ');

Repeat

ch:=ReadKey;

tap:=tap+[ch];

Writeln(ch);

Until not(ch in ['a'..'z']);

Writeln('Cac phan tu cua tap hop la:');

For ch:='a' to 'z' do If ch in tap then Write(ch, ' ');

Writeln;

Writeln('Ban muon bo cac phan tu nao khoi tap hop:');

Repeat

ch:=ReadKey;

tap:=tap-[ch];

Writeln(ch);

Until not(ch in ['a'..'z']);

Writeln('Cac phan tu con lai cua tap hop la:');

For ch:='a' to 'z' do If ch in tap then Write(ch, ' ');

Readln;

END.

Bài 4. Mô phỏng đồ thị

3. Ma trận kề của đồ thị:

```
0 1 1 0 1
1 0 1 1 0
1 1 0 1 0
0 1 1 0 1
1 0 0 1 0
```

5. Program P20405; {Tạo ma trận kề của đồ thị}

Uses Crt;

Const gn = 'P20405.TXT';

MN = 64;

huong: array [1..8,1..2] of integer = ((-1,-1),(-1,0),
(-1,1),(0,-1),(0,1),(1,-),(1,0),(1,1));

Type mang2 = array [1..MN,1..MN] of byte;

Var a: mang2;

g: text;

Procedure Tinh(m,n: integer);

Var i: integer;

Begin

For i:=1 to 8 do

If (m+huong[i,1]>0)and(m+huong[i,1]<=8) then

If (n+huong[i,2]>0)and(n+huong[i,2]<=8) then

a[(m-1)*8+n,(m+huong[i,1]-1)*8+n+huong[i,2]]:=1;

End;

Procedure Xuly;

Var i,j: integer;

Begin

Assign(g,gn); Rewrite(g);

FillChar(a,Sizeof(a),0);

For i:=1 to 8 do

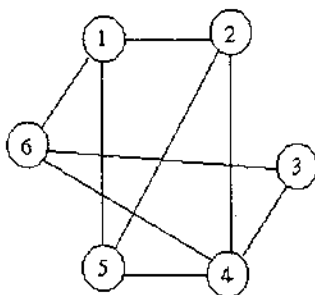
For j:=1 to 8 do Tinh(i,j);

```

For i:=1 to 64 do
begin
  For j:=1 to 64 do Write(g,a[i,j]:2);
  Writeln(g);
end;
Close(g);
End;
BEGIN
  Xuly;
END.

```

7. Đồ thị:



9. Ma trận kề của đồ thị:

	A	B	C	D	E
A	0	1	0	1	0
B	0	0	0	1	0
C	1	1	0	0	0
D	0	0	1	0	1
E	0	1	0	0	0

10. Program P20410; {Tạo danh sách kề của đồ thị}

Uses Crt;

Const MN = 100;

fn = 'P20410.INP';

gn = 'P20410.OUT';

Type

mang2=array [1..MN,1..MN] of byte;

Var a: mang2;

n: integer;

f,g: text;

Procedure Nhap;

Var i,j: integer;

Begin

Assign(f,fn); Reset(f);

Readln(f,n);

For i:=1 to n do

For j:=1 to n do Read(f,a[i,j]);

Close(f);

End;

Procedure Viet;

Var i,j: integer;

Begin

Assign(g,gn); Rewrite(g);

Writeln(g,n,' : ');

For i:=1 to n do

begin

For j:=1 to n do

If a[i,j]=1 then Write(g,j,' ');

Writeln(g);

end;

Close(g);

End;

BEGIN

ClrScr;

Nhap;

Viet;

END.

11. Program P20411; {Tao ma tran ke cua do thi}

Uses Crt;

Const MN = 100;

fn = 'P20411.INP';

gn = 'P20411.OUT';

Type

mang2 = array [1..MN,1..MN] of byte;

Var a: mang2;

n: integer;

f,g: text;

Procedure Viet;

Var i,j: integer;

Begin

Assign(g,gn); Rewrite(g);

Writeln(g,n);

For i:=1 to n do

begin

For j:=1 to n do Write(g,a[i,j]:2);

Writeln(g);

end;

Close(g);

End;

Procedure Nhap;

Var i,j: integer;

Begin

Assign(f,fn); Reset(f);

Readln(f,n);

FillChar(a,SizeOf(a),0);

For i:=1 to n do

begin

While not SeekEoln(f) do begin Read(f,j); a[i,j]:=1; end;

```

        Readln(f);
    end;
    Close(f);
End;

```

```

BEGIN
    ClrScr;
    Nhap;
    Viet;
END.

```

13. Program P20413; {Tim phan lien thong chua dinh S}

```

Uses Crt;
Const  MN = 100;
        fn = 'P20413.INP';
        gn = 'P20413.OUT';
Type   mang2=array [1..MN,1..MN] of byte;
        mang1=array [1..MN] of byte;
Var    a: mang2;
        st,kt: mang1;
        n,s,sst: integer;
        f,g: text;

```

```

Procedure Nhap;
Var i,j: integer;
Begin
    Assign(f,fn); Reset(f);
    Readln(f,n);
    For i:=1 to n do
        For j:=1 to n do Read(f,a[i,j]);
    Close(f);
    Write('s = '); Readln(s);
End;

```

```

Procedure Init;
Begin
  FillChar(kt,SizeOf(kt),0);
  sst:=1; st[sst]:=s; kt[s]:=1;
End;

Procedure Viet;
Var d,i: integer;
Begin
  Assign(g,gn); Rewrite(g);
  Writeln(g,'Buoc 1');
  Writeln(g,'khởi tạo Queue');
  Writeln(g,'{',s,'}');
  Writeln(g);
  d:=0;
  While d<sst do
    begin
      Inc(d); s:=st[d];
      Writeln(g,'Buoc ',d+1);
      Writeln(g,'Xét đỉnh ',s);
      For i:=1 to n do
        If (a[s,i]=1)and(kt[i]=0) then
          begin
            Inc(sst);
            st[sst]:=i;
            kt[i]:=1;
          end;
      Write(g,'{ ');
      For i:=1 to d do Write(g,st[i],' ');
      Write(g,' | ');
      For i:=d+1 to sst do Write(g,st[i],' ');
      Writeln(g,'}'); Writeln(g);
    end;
  Close(g);

```

End;

BEGIN

Clrscr;

Nhap;

Init;

Viet;

END.

15. Program P20415; {Tim duong di tu dinh S den dinh T}

Uses Crt;

Const MN = 100;

fn = 'P20415.inp';

gn = 'P20415.out';

Type

mang2 = array [1..MN,1..MN] of byte;

mang1 = array [1..MN] of byte;

Var a: mang2;

st,kt,tu,duong: mang1;

n,s,t,sst,sd: integer;

f,g: text;

Procedure Nhap;

Var i,j: integer;

Begin

Assign(f,fn); Reset(f);

Readln(f,n);

For i:=1 to n do

For j:=1 to n do Read(f,a[i,j]);

Close(f);

Write('s = '); Readln(s);

Write('t = '); Readln(t);

End;

```

Procedure Init;
Begin
  FillChar(kt,Sizeof(kt),0);
  FillChar(tu,Sizeof(tu),0);
  sst:=1; st[sst]:=s; kt[s]:=1;
End;

```

```

Procedure Timduong;
Var i: integer;
Begin
  sd:=1; duong[1]:=t;
  While tu[duong[sd]]<>0 do
    begin
      Inc(sd);
      duong[sd]:=tu[duong[sd-1]];
    end;
  For i:=sd downto 1 do Write(g,duong[i],' ');
  Readln;
End;

```

```

Procedure Viet;
Var d,i: integer;
Begin
  Assign(g,gn); Rewrite(g);
  Writeln(g,'Buoc 1');
  Writeln(g,'khởi tạo Queue');
  Writeln(g,'{',s,'}');
  Writeln(g);
  d:=0;
  While (d<sst)and(tu[t]=0) do
    begin
      Inc(d); s:=st[d];
      Writeln(g,'Buoc ',d+1);
      Writeln(g,'Xét định ',s);
    end;

```

```

For i:=1 to n do
  If (a[s,i]=1)and(kt[i]=0) then
    begin
      Inc(sst);
      st[sst]:=i;
      kt[i]:=1;
      tu[i]:=s;
    end;
  Write(g,'{ ');
  For i:=1 to d do Write(g,st[i]:4); Write(g,'| ');
  For i:=d+1 to sst do Write(g,st[i]:4); Writeln(g,'}');
  Write(g,'{ ');
  For i:=1 to d do If tu[st[i]]<>0 then Write(g,tu[st[i]]:4)
    Else Write(g,' ');
  Write(g,'| ');
  For i:=d+1 to sst do If tu[st[i]]<>0 then Write(g,tu[st[i]]:4)
    Else Write(g,' ');
  Writeln(g,'}'); Writeln(g);
end;
If tu[t]=0 then Write(g,' Vo nghiem ');
Else Timduong;
Readln;
Close(g);
End;

BEGIN
  Clrscr;
  Nhap;
  Init;
  Viet;
END.

```

16. Program P20416; {Tim duong di cua con ma tren ban co}
 Uses Crt;

```

Const MN = 64;
      gn = 'P20416.OUT';
      h: array [1..8,1..2] of integer = ((-2,-1),(-2,1),(-1,-2),(-
1,2),(1,-2),(1,2),(2,-1),(2,1));
Type mangl=array [1..MN] of byte;
Var ng,kt,st: mangl;
      sst,s,t: integer;
      g: text;

```

```

Procedure Nhap;
Begin
  Write('s = '); Readln(s);
  Write('t = '); Readln(t);
end;

```

```

Procedure Init;
Begin
  FillChar(kt,SizeOf(kt),0);
  sst:=1; st[1]:=s; kt[s]:=65;
End;

```

```

Procedure Loang(d: integer);
Var x,y,i: integer;
Begin
  y:=d mod 8; x:=d div 8;
  If y=0 then y:=8 Else Inc(x);
  For i:=1 to 8 do
    If (x+h[i,1]>0)and(x+h[i,1]<=8) then
      If (y+h[i,2]>0)and(y+h[i,2]<=8) then
        If kt[(x+h[i,1]-1)*8+y+h[i,2]]=0 then
          begin
            Inc(sst); st[sst]:=(x+h[i,1]-1)*8+y+h[i,2];
            kt[(x+h[i,1]-1)*8+y+h[i,2]]:=(x-1)*8+y;
          end;

```

End;

Procedure Xuly;

Var d: integer;

Begin

d:=1;

While (d<=sst)and(kt[t]=0) do

begin Loang(st[d]); Inc(d); end;

End;

Procedure Viet;

Var d,i: integer;

Begin

Assign(g,gn); Rewrite(g);

If kt[t]=0 then Write(g,'Vo nghiem')

Else

begin

d:=1; ng[d]:=t;

While kt[ng[d]]<>65 do

begin

Inc(d);

ng[d]:=kt[ng[d-1]];

end;

For i:=d downto 1 do Write(g,ng[i], ' ');

end;

Close(g);

End;

BEGIN

Clrscr;

Nhap;

Init;

Xuly;

Viet;

END.

Bài 5. Kiểu bản ghi

1. Type

```
hocsinh = record
    hoten: string[26];
    tuoi: byte;
end;
giaovien = array[1..10] of string[26];
lop = record
    ten: string[4];
    gv: giaovien;
    hs: hocsinh;
end;
```

```
Var class: lop;
```

3. Program P20503;

```
Uses Crt;
```

```
Type kieu = record
    hoten: string[26];
    tuoi: byte;
end;
```

```
Var i: integer;
    a: array [1..20] of kieu;
```

```
BEGIN
```

```
Writeln('Nhap vao f. Thoi nhap khi i=20.');
```

```
For i:=1 to 20 do With a[i] do
```

```
begin
```

```
    Write('Hoten: '); Readln(hoten);
```

```
    GotoXY(40,WhereY-1);
```

```
    Write('Tuoi: '); Readln(a[i].tuoi);
```

```
end;
```

```
END.
```

5. Program P20505; {Nhap danh sach hoc sinh tu ban phim}

Uses Crt;

Const n = 15;

Type Danhsach = record

 holot: string[25];

 ten: string[10];

 tuoi: 0..99;

 lop: string[3];

end;

Var ds: array [1..n] of Danhsach;

 i: byte;

BEGIN

 ClrScr;

 Writeln('Hay nhap danh sach hoc sinh : ');

 Writeln;

 For i:=1 to n do

 begin

 Writeln('Thong tin hoc sinh thu ',i);

 Write('Cho ho lot : '); Readln(ds[i].holot);

 Write('Cho ten : '); Readln(ds[i].ten);

 Write('Cho tuoi : '); Readln(ds[i].tuoi);

 Write('Cho lop : '); Readln(ds[i].lop);

 Writeln;

 end;

 Writeln('Danh sach hoc sinh :');

 For i:=1 to n do

 With ds[i] do Writeln(holot:20,ten:10,tuoi:4,lop:5);

 Writeln;

 Write('Bam Enter de ket thuc...');

 Readln;

END.

7. Program P20507; {Nhap danh sach vao File ban ghi}

```

Uses Crt;
Const n = 20;
Type Danhsach = record
    holot: string[25];
    ten: string[10];
    tuoi: 0..99;
    lop: string[3];
end;
Var ds: Danhsach;
    i: byte;
    f: file of Danhsach;

BEGIN
    ClrScr;
    Assign(f,'P20507.DAT'); Rewrite(f);
    Writeln('Hay nhap danh sach hoc sinh : ');
    Writeln;
    For i:=1 to n do
        begin
            Writeln('Thong tin hoc sinh thu ',i);
            Write('Cho ho lot : '); Readln(ds.holot);
            Write('Cho ten   : '); Readln(ds.ten);
            Write('Cho tuoi   : '); Readln(ds.tuoi);
            Write('Cho lop    : '); Readln(ds.lop);
            Write(f,ds);
            Writeln;
        end;
    Close(f);
    Writeln;
    Write('Bam Enter de ket thuc...');
    Readln;
END.

```

9. Program P20509; {Hien thi danh sach ten sap theo thu tu}

Uses Crt;
 Const n = 20;
 Type Danhsach = record
 holot: string[25];
 ten: string[10];
 tuoi: 0..99;
 lop: string[3];
 end;
 Var ds: array [1..20] of Danhsach;
 i,j: byte;
 f: file of Danhsach;

Procedure Doi(i,j: byte);
 Var tg: Danhsach;
 Begin
 tg:=ds[i];
 ds[i]:=ds[j];
 ds[j]:=tg;
 End;

BEGIN
 ClrScr;
 Writeln('Nhap danh sach hoc sinh tu file P20509.DAT:');
 Writeln;
 Assign(f,'P20509.DAT'); Reset(f);
 For i:=1 to n do Read(f,ds[i]);
 Close(f);
 For i:=1 to n-1 do
 For j:=i+1 to n do
 begin
 If (ds[i].ten>ds[j].ten) then Doi(i,j)
 Else If (ds[i].ten=ds[j].ten)and(ds[i].holot>ds[j].holot)
 then Doi(i,j);
 end;
 end;

Writeln('Danh sach hoc sinh:');
 For i:=1 to n do
 With ds[i] do Writeln(holot:20,ten:11,tuoi:4,lop:5);
 Writeln;
 Write('Bam Enter de ket thuc...');
 Readln;
 END.

11. Program P20511; {Hien thi danh sach lop 11A}

Uses Crt;

Type Danhsach = record
 holot: string[25];
 ten: string[10];
 tuoi: 0..99;
 lop: string[3];
 end;

Var ds: Danhsach;
 i,d: byte;
 f: file of Danhsach;

BEGIN

 ClrScr;

 Writeln('Danh sach hoc sinh tu file P20511.DAT');

 Writeln;

 Assign(f,'P20511.DAT'); Reset(f);

 d:=0;

 While not Eof(f) do

 begin

 Read(f,ds);

 With ds do

 If (ds.lop='11A') then

 begin

 Writeln(holot:20,ten:10,tuoi:4,lop:5);

 Inc(d);

 end;

end;
 If (d=0) then Writeln('Danh sach khong chua hoc sinh lop 11A.');

Close(f);
 Writeln;
 Write('Bam Enter de ket thuc...');

Readln;
 END.

13. Program P20513; {Doi kieu ban ghi}
 Uses Crt;
 Type

Danhsach1 = record
 holot: string[25];
 ten: string[10];
 tuoi: 0..99;
 lop: string[3];
 end;
 Danhsach2 = record
 hoten: string[35];
 tuoi: byte;
 khai: byte;
 lop: char;
 end;

Var ds1: Danhsach1;
 ds2: Danhsach2;
 f1: file of Danhsach1;
 f2: file of Danhsach2;
 c: integer;

BEGIN
 ClrScr;
 Writeln('Ghi tu file P20513.DAT sang P20513N.DAT:');
 Writeln;

```

Assign(f1,'P20513.DAT'); Reset(f1);
Assign(f2,'P20513N.DAT'); Rewrite(f2);
While not Eof(f1) do
    begin
        Read(f1,ds1);
        With ds1 do
            begin
                ds2.hoten:=holot+ten;
                val(copy(lop,1,2),ds2.khoi,c);
                ds2.tuoi:=tuoi;
                ds2.lop:=UpCase(lop[3]);
                Write(f2,ds2);
            end;
        end;
    Close(f1); Close(f2);
    Writeln;
    Writeln('Bam Enter de ket thuc!');
    Readln;
END.

```

15. Program P20515; {Tim kiem tren ban ghi}

Uses Crt;

Type

```

    Danhsach=record
        Tacgia: string[26];
        Tensach: string[40];
        NamXB: integer;
    end;

```

Var ds: array [1..100] of Danhsach;

n: integer;

f: file of Danhsach;

M: Danhsach;

Procedure Nhap;

```

Begin
n:=0;
Assign(f,'P20515.DAT'); Reset(f);
While not Eof(f) do
    begin
        Inc(n);
        Read(f,ds[n]);
    end;
Close(f);
End;

```

```

Procedure TheoTG;
Var tacgia: string;
    namXB: integer;
    i: integer;
Begin
Write('Cho ten tac gia : '); Readln(M.tacgia);
Write('Cho nam xuat ban : '); Readln(M.NamXB);
i:=1;
While
(i<=n)and(((ds[i].tacgia<>tacgia)or(ds[i].namXB<>namXB)) do
i:=i+1;
If (i>n) then Writeln('Khong tim duoc')
Else Writeln(ds[i].tacgia:28,ds[i].tensach:42,ds[i].namXB:6);
End;

```

```

Procedure TheoTS;
Var ten: string;
    i: integer;
Begin
Write('Cho ten sach : '); Readln(ten);
i:=1;
While (i<=n)and(ds[i].tensach<>ten) do i:=i+1;
If (i>n) then Writeln('Khong tim duoc')

```


Else Writeln(ds[i].tacgia:28,ds[i].tensach:42,ds[i].namXB:6);
End;

Procedure Timkiem;

Var c:char;

Begin

Writeln('1. Tim kiem theo tac gia va nam xuat ban.');

Writeln('2. Tim kiem theo ten sach');

Writeln;

Write('Ban chon [1/2] : ');

Repeat

 c:=Readkey;

Until pos(c,'12')>0;

Writeln(c);

If c='1' then TheoTG Else TheoTS;

End;

BEGIN

 ClrScr;

 Nhap;

 Timkiem;

 Write('Bam Enter de ket thuc...');

 Readln;

END.

Bài 6. Kiểu file

1. Các khai báo đều đúng nhưng chỉ có khai báo thứ nhất cho phép truy cập dữ liệu có kiểu là kiểu của file (kiểu byte), còn các khai báo khác thì không được.

3. Khai báo thứ nhất cho phép truy cập file với các biến có kiểu là a còn khai báo thứ hai không cho phép truy cập file với các biến có kiểu array[1..10] of integer.

5. Program P20605; {Tao mot tep cac so nguyen}

Uses Crt;

Var i: Integer;

f: file of Integer;

BEGIN

Assign(f,'Thu.dat'); Rewrite(f);

For i:=1 to 1000 do Write(f,i);

Close(f);

END.

7. Program P20607; {Ghi va doc file of String}

Uses Crt;

Const INP='Thu.dat';

Type String10 = String[10];

Procedure Ghi;

Var f: file of String10;

S: String10;

Begin

Clrscr;

Assign(f,INP); Rewrite(f);

Writeln('Nhap vao f. Thoi nhap khi S="" (go Enter)!');

Readln(S);

While (S<>") do

begin

```

        Write(f,S);
        Readln(S);
    end;
    Close(f);
End;

Procedure Doc;
Var f: file of String10;
    S: String10;
Begin
    Clrscr;
    Assign(f,INP); Reset(f);
    While Not Eof(f) Do
        begin
            Read(f,S);
            Writeln(S);
            If WhereY=25 then
                begin
                    Write('Press Enter to continue..');
                    Readln; Clrscr;
                end;
        end;
    Close(f);
    Readln;
End;

BEGIN
    Ghi;
    Doc;
END.

```

9. Hãy chạy thử chương trình.

11. Program P20611; {ghi de vao tep cac ban ghi vao vi tri tuy chon}

Uses Crt;

Const inp = 'Hocsinh.dat';

Type Hocsinh = Record

 Ten : String[30];

 Tuoi: Byte;

End;

Var f : file of Hocsinh;

 Hs: Hocsinh;

 k : integer;

BEGIN

 Assign(f,inp); Reset(f);

 Write('Ho va ten: '); Readln(Hs.Ten);

 Write('Tuoi : '); Readln(Hs.Tuoi);

 Write('Vi tri de: '); Readln(k);

 Seek(f,k); Write(f,Hs);

 Close(f);

END.

13. Program P20613; {chen vao tep cac ban ghi vao vi tri tuy chon}

Uses Crt;

Const inp = 'Hocsinh.dat';

Type Hocsinh = Record

 Ten : String[30];

 Tuoi: Byte;

End;

Var f : file of Hocsinh;

 Hs,Hs1: Hocsinh;

 k : integer;

BEGIN

```

Assign(f,inp); Reset(f);
Write('Ho va ten: '); Readln(Hs1.Ten);
Write('Tuoi   : '); Readln(Hs1.Tuoi);
Write('Vi tri chen: '); Readln(k);
Seek(f,k);
While not Eof(f) do
    begin
        Read(f,Hs);
        Seek(f,k);
        Write(f,Hs1);
        Hs1:=Hs;
        Inc(k);
    end;
Write(f,Hs1);
Close(f);
END.

```

```

15. Program P20615;
Uses Crt;
Const inp = 'Hocsinh.dat';
Type Hocsinh = Record
    Ten : String[30];
    Tuoi: Byte;
End;
Var f : file of Hocsinh;
    Hs,Hs1: Hocsinh;
    i,k : integer;
    c: char;

```

```

Procedure Xem; {Xem danh sach cac ban ghi}
Begin
    ClrScr;
    Seek(f,0); k:=-1;
    While not Eof(f) do

```

```

begin
    Read(f,Hs); Inc(k);
    Writeln(k:3,Hs.Ten:26,Hs.Tuoi:3);
    If WhereX=25 then
        begin
            Write('Nhan Enter tiep tuc...');
            Readln; Clrscr;
        end;
    end;
Readln;
End;

Procedure Xoa; {Xoa mot ban ghi o vi tri tuy chon}
Const Tg = 'Trunggian.d';
Var f2: file of Hocsinh;
Begin
    ClrScr;
    Write('Thu tu ban ghi muon xoa: '); Readln(k);
    Assign(f2,Tg); Rewrite(f2);
    Seek(f,0);
    For i:=0 to k-1 do
        begin
            Read(f,Hs);
            Write(f2,Hs);
        end;
    Read(f,Hs); {bo qua ban ghi nay}
    While not Eof(f) do
        begin
            Read(f,Hs);
            Write(f2,Hs);
        end;
    Close(f); Close(f2);
    Erase(f); Rename(f2,inp);
    Assign(f,inp); Reset(f);
End;

```

```

Procedure Sua; {Sua mot ban ghi o vi tri tuy chon}
Begin
  ClrScr;
  Write('Vi tri can sua: '); Readln(k);
  Seek(f,k); Read(f,Hs);
  Writeln(Hs.Ten:25,Hs.Tuoi:3);
  Write('Ho va ten moi: '); Readln(Hs.Ten);
  Write('Tuoi      : '); Readln(Hs.Tuoi);
  Seek(f,k); Write(f,Hs);
End;

```

```

Procedure Them; {chen mot ban ghi vao vi tri tuy chon}
Begin
  ClrScr;
  Write('Ho va ten : '); Readln(Hs1.Ten);
  Write('Tuoi      : '); Readln(Hs1.Tuoi);
  Write('Vi tri chen: '); Readln(k);
  If k<=FileSize(f) then Seek(f,k) Else Seek(f,FileSize(f));
  While not Eof(f) do
    begin
      Read(f,Hs);
      Seek(f,k);
      Write(f,Hs1);
      Hs1:=Hs; Inc(k);
    end;
  Write(f,Hs1);
End;

```

```

Procedure Ve_Menu;
Begin
  Clrscr;
  Writeln('X. Xem toan bo danh sach. ');
  Writeln('S. Sua chua mot ban ghi. ');

```

```

Writeln('H. Xoa mot ban ghi. ');
Writeln('T. Them mot ban ghi vao danh sach. ');
Writeln('Q. Thoat khoi chuong trinh. ');
Write('Lua chon cua ban: '); c:=Readkey;
c:= Uppercase(c); Write(c);
End;

```

```

Procedure Xuli;
Begin
  Case c of
    'X': Xem;
    'S': Sua;
    'H': Xoa;
    'T': Them;
    'Q': begin Close(f); Halt; end;
  End;
End;

```

```

BEGIN
  Assign(f,inp); Reset(f);
  Repeat
    Ve_menu;
    Xuli;
  Until false;
END.

```


Bài 7. Kiểu con trỏ

1. Program P20701;

{ Khai bao bien kieu con tro Pointer }

Type

BytePtr = ^Byte;

WordPtr = ^Word;

IdentPtr = ^IdentRec;

IdentRec = record

Ident: string[15];

RefCount: Word;

Next: IdentPtr;

end;

3. Program P20703;

Uses Crt;

Type

TFriendRec = record

Name: string[30];

Age : Byte;

end;

Var p: pointer;

BEGIN

Clrscr;

If MaxAvail < SizeOf(TFriendRec) then Writeln('Not enough
memory')

Else

begin

GetMem(p, SizeOf(TFriendRec));

FreeMem(p, SizeOf(TFriendRec));

end;

Readln;

END.

5. Program P20705;

Type kdl = integer;

Var i: ^kdl;

 a: kdl;

BEGIN

 i:=Addr(a);

 a:=100;

 Writeln(i^);

 i^:=200;

 Writeln(a);

 Readln;

END.

7. Chạy thử chương trình.

9. Program P20709; {danh sach ten moc noi 2 chieu}

Uses Crt;

Type ptr = ^rec;

 rec = record

 previous: ptr;

 name: string[20];

 next: ptr;

 end;

Var p,l,r: ptr;

Procedure Nhap;

Begin

 ClrScr;

 New(p); l:=p;

 Write('Ten: '); Readln(p^.name); p^.previous:=nil;

 While p^.name<>" do

 begin

 New(p^.next);

 p^.next^.previous:=p;

```

        p:=p^.next;
        Write('Ten: '); Readln(p^.name);
    end;
    p^.next:=nil; r:=p;
End;

```

```

Procedure Lietke_Xuoi;
Begin
    Writeln('Liet ke xuai danh sach da nhap:');
    While l^.next<>nil do
        begin
            Writeln(l^.name);
            l:=l^.next;
        end;
    Readln;
End;

```

```

Procedure Lietke_Nguoc;
Begin
    Writeln('Liet ke nguoc danh sach da nhap:');
    Repeat
        Writeln(r^.name);
        r:=r^.previous;
    Until r=nil;
    Readln;
End;

```

```

BEGIN
    Nhap;
    Lietke_Xuoi;
    Lietke_Nguoc;
END.

```

11. Program P20711; {Go bo ban ghi khoi danh sach}

```

Uses Crt;
Type ptr = ^rec;
      rec = record
          name: string[20];
          next: ptr;
      end;
Var k: integer;
    p,l: ptr;

```

```

Procedure Nhap;
Begin
    ClrScr;
    New(p); l:=p;
    Write('Ten: '); Readln(p^.name);
    Repeat
        New(p^.next);
        p:=p^.next;
        Write('Ten: '); Readln(p^.name);
    Until p^.name="";
    p^.next:=nil;
    Write('Vi tri ban ghi can go bo: '); Readln(k);
End;

```

```

Procedure Gobo;
Var i: integer; q: Ptr;
Begin
    p:=l;
    For i:=1 to k do p:=p^.next;    {Tim vi tri cuoi}
    q:=p;
    p:=l;
    For i:=3 to k do p:=p^.next;    {Tim vi tri dau}
    If k=1 then l:=q
    Else p^.next:=q;
End;

```

```

Procedure In_kq;
Begin
  While (l^.next <> nil) do
    begin
      Writeln(l^.name);
      l:=l^.next;
    end;
  Readln;
End;

```

```

BEGIN
  Nhap;
  Gobo;
  In_kq;
END.

```

13. Program P20713; {Chen them ban ghi vao danh sach}

```

Uses Crt;
Const inp = 'P20713.txt';
Type ptr = ^rec;
      rec = record
        name: string[20];
        next: ptr;
      end;
Var f: text;
    k: integer;
    p,s,l: ptr;

```

```

Procedure Nhap;
Begin
  Assign(f,inp); Reset(f);
  New(p); l:=p;
  While not Eof(f) do

```

```

begin
    Readln(f,p^.name);
    New(p^.next);
    p:=p^.next;
end;
p^.next:=nil;
Close(f);

New(s);
Clrscr;
Writeln('Nhap ban ghi can chen: ');
Write('Ten: '); Readln(s^.name);
Write('Vi tri can chen: '); Readln(k);
End;

```

```

Procedure Chen_rec;
Var i: integer;

```

```

    Procedure Cat(Var L: ptr);
    Begin
        s^.next:=l;
        l:=s;
    end;

```

```

Begin
    p:=l;
    For i:=3 to k do p:=p^.next;    {Tim vi tri}
    If k>1 then Cat(p^.next)
    Else Cat(l); {Cat - Noi}
End;

```

```

Procedure In_kq;
Begin
    While (l^.next<>nil) do

```

```

    begin
        Writeln(l^.name);
        l:=l^.next;
    end;
Readln;
End;

BEGIN
    Nhap;
    Chen_rec;
    In_kq;
END.

```

15. Program P20715; {Sap xep cac ban ghi trong danh sach theo Name}

```

Uses Crt;
Const inp = 'P20715.txt';
Type ptr=^rec;
    rec=record
        name: string[20];
        next: ptr;
    end;
Var f: text;
    n,i,j: integer;
    p,l: ptr;
    teni,tenj: string;

```

```

Procedure Nhap;
Begin
    Assign(f,inp); Reset(f);
    New(p); l:=p; n:=0;
    While not Eof(f) do
        begin
            Readln(f,p^.name);

```

```

        New(p^.next);
        p:=p^.next;
        Inc(n);
    end;
    p^.next:=nil;
    Close(f);
End;

```

```

Procedure Doicho(i,j: integer);
Var k: integer;
Begin
    p:=l;
    For k:=2 to i do p:=p^.next;
    p^.name:=tenj;
    p:=l;
    For k:=2 to j do p:=p^.next;
    p^.name:=teni;
End;

```

```

Procedure Sapxep;
Var k: integer;
Begin
    For i:=1 to n-1 do
        For j:=i+1 to n do
            begin
                p:=l; For k:=2 to i do p:=p^.next; teni:=p^.name;
                p:=l; For k:=2 to j do p:=p^.next; tenj:=p^.name;
                If teni>tenj then Doicho(i,j);
            end;
        end;
    End;

```

```

Procedure In_kq;
Begin
    While (l^.next<>nil) do

```



```

begin
    Writeln(l^.name);
    l:=l^.next;
end;
Readln;
End;

```

```

BEGIN
    Clrscr;
    Nhap;
    Sapxep;
    In_kq;
END.

```

17. Program P20717; {dung bien bo nho}

```

Uses Crt, Graph;
Const X=400; Y=200; W=100; H=20; G=15;
Var Gd,Gm,i,j: Integer;
    P: Pointer;
    Size: Word;

```

```

Procedure VeXetang;
Begin
    SetBkColor(0);
    SetFillStyle (1,7);
    Bar(X,Y,X+W,Y+W);
    SetColor(8);
    Line(X,Y+H,X+W,Y+H);
    Line(X,Y+W-H,X+W,Y+W-H);
    Circle(X+W div 2-5,Y+W div 2,W div 2-H-5);
    Circle(X+W div 2-5,Y+W div 2,W div 2-H-13);
    SetFillStyle (1,8);
    Bar(X,Y+H,X+G,Y+W-H);
    FloodFill(X+W div 2-5,Y+H+10,8);

```

SetFillStyle (1,7);
Bar(X-5,Y+W div 2-2,X+H+3,Y+W div 2+2);
Readln;
End;

BEGIN
Gd:=Detect;
InitGraph(Gd,Gm,' ');
VeXetang;
Size:=ImageSize(X,Y,X+W,Y+W);
GetMem(P,Size);
GetImage(X,Y,X+W,Y+W,P^);
i:=0; j:=200;
PutImage(i,j,P^,NormalPut);
Readln;
CloseGraph;
END.

Bài 8. Vẽ đồ thị hàm số

1. Program P20801; {vẽ đồ thị hàm $y = x/2$ }

Uses Crt, Graph;

Const i0=300; j0=250; E=50;

Var Gd, Gm, i, j, c: Integer;

 x, y: real;

 S: string;

Function f(x: real): real;

Begin

 f:=x/2;

End;

BEGIN

 Gd:=Detect;

 InitGraph(Gd, Gm, '');

 Line(i0, 50, i0, 400);

 Line(50, j0, 600, j0);

 OutTextXY(600-3, j0+5, 'x');

 OutTextXY(i0+5, 50, 'y');

 For i:=51 to 599 do

 If (i mod E=0) then

 begin

 Bar (i-1, j0-1, i+1, j0+1);

 x:=(i-i0)/E;

 Str(x:0:0, S);

 If x > 0 then OutTextXY(i-3, j0+5, S);

 end;

 For j:=51 to 399 do

 If (j mod E=0) then

 begin

 Bar (i0-1, j-1, i0+1, j+1);

 y:=-(j-j0)/E;

```

        Str(y:0:0,S);
        If  $y < 0$  then OutTextXY(i0+5,j,S);
    end;
For i:=70 to 580 do
    begin
        x:=(i-i0)/E;
        y:=f(x);
        j:=round(-y*e+j0);
        If (j>70)and(j<380) then PutPixel(i,j,2);
    end;
Readln;
CloseGraph;
END.

```

3. Program P20803; {ve do thi ham $y = \text{sqr}(x)/2$ }

Uses Crt,Graph;

Const i0=300; j0=250; E=50;

Var Gd,Gm,i,j,c: Integer;

x,y: real;

S: string;

Function f(x: real): real;

Begin

f:=Sqr(x)/2;

End;

BEGIN

Gd:=Detect;

InitGraph(Gd, Gm, '');

Line(i0,50,i0,400);

Line(50,j0,600,j0);

OutTextXY(600-3,j0+5,'x');

OutTextXY(i0+5,50,'y');

For i:=51 to 599 do

```

if (i mod E=0) then
  begin
    Bar (i-1,j0-1,i+1,j0+1);
    x:=(i-i0)/E;
    Str(x:0:0,S);
    If x<>0 then OutTextXY(i-3,j0+5,S);
  end;
For j:=51 to 399 do
  If (j mod E=0) then
    begin
      Bar (i0-1,j-1,i0+1,j+1);
      y:=(j-j0)/E;
      Str(y:0:0,S);
      If y<>0 then OutTextXY(i0+5,j,S);
    end;
For i:=70 to 580 do
  begin
    x:=(i-i0)/E;
    y:=f(x);
    j:=round(-y*e+j0);
    If (j>70)and(j<380) then PutPixel(i,j,2);
  end;
Readln;
CloseGraph;
END.

```

5. Program P20805; {ve do thi ham $y = \sin x$ }

Uses Crt,Graph;

Const i0=300; j0=250; E=50;

Var Gd,Gm,i,j,c: Integer;

x,y: real;

S: string;

Function f(x: real): real;

Begin

```
f:=sin(x);
End;
```

```
BEGIN
```

```
  Gd:=Detect;
  InitGraph(Gd, Gm, '');
  Line(i0,50,i0,400);
  Line(50,j0,600,j0);
  OutTextXY(i0+5,j0+5,'0');
  OutTextXY(600-3,j0+5,'x');
  OutTextXY(i0+5,50,'y');
  For i:=51 to 599 do
    If (i mod E=0) then
      begin
        Bar (i-1,j0-1,i+1,j0+1);
        x:=(i-i0)/E;
        Str(x:0:0,S);
        If x<>0 then OutTextXY(i-3,j0+5,S);
      end;
```

```
  For j:=51 to 399 do
    If (j mod E=0) then
      begin
        Bar (i0-1,j-1,i0+1,j+1);
        y:=(j-j0)/E;
        Str(y:0:0,S);
        If y<>0 then OutTextXY(i0+5,j,S);
      end;
```

```
  For i:=70 to 580 do
    begin
      x:=(i-i0)/E;
      y:=f(x);
      j:=round(-y*e+j0);
      If (j>70)and(j<380) then PutPixel(i,j,2);
```

```

    end;
  Readln;
  CloseGraph;
END.

```

7. Program P20807; {ve do thi ham $y = 1/x$ }

Uses Crt, Graph;

Const i0=300; j0=250; E=50;

Var Gd, Gm, i, j, il, jl: Integer;

 x, y: real;

 S: string;

Function f(x: real): real;

Begin

 f:=1/x;

End;

BEGIN

 Gd:=Detect;

 InitGraph(Gd, Gm, '');

 Line(i0, 50, i0, 400);

 Line(50, j0, 600, j0);

 OutTextXY(i0+5, j0+5, '0');

 OutTextXY(600-3, j0+5, 'x');

 OutTextXY(i0+5, 50, 'y');

 For i:=51 to 599 do

 If (i mod E=0) then

 begin

 Bar (i-1, j0-1, i+1, j0+1);

 x:=(i-i0)/E;

 Str(x:0:0, S);

 If x <> 0 then OutTextXY(i-3, j0+5, S);

 end;

 For j:=51 to 399 do

 If (j mod E=0) then

139,899,7
1,632,664

```

begin
  Bar (i0-1,j-1,i0+1,j+1);
  y:=(j-j0)/E;
  Str(y:0:0,S);
  If y<0 then OutTextXY(i0+5,j,S);
end;
For i:=70 to 580 do
  If i<i0 then
    begin
      x:=(i-i0)/E;
      y:=f(x);
      j:=round(-y*e+j0);
      SetColor(2);
      If (i<70)and(j>70)and(j<380) then Line(i,j,il,jl);
      il:=i; jl:=j;
    end;
  Readln;
  CloseGraph;
END.

```

9. Program P20809; {ve do thi ham $y = (\text{sqr}(x)-x+1)/(x-1)$ }

Uses Crt,Graph;

Const i0=300; j0=250; E=30;

Var Gd,Gm,i,j,il,jl: Integer;

x,y: real;

S: string;

Function f(x: real): real;

Begin

f:=(sqr(x)-x+1)/(x-1);

End;

BEGIN

Gd:=Detect;

InitGraph(Gd, Gm, '');


```

Line(i0,50,i0,400); {Oy}
Line(50,j0,600,j0); {Ox}
OutTextXY(i0+5,j0+5,'0');
OutTextXY(600-3,j0+5,'x');
OutTextXY(i0+5,50,'y');
For i:=51 to 599 do
  If ((i-i0) mod E=0) then
    begin
      Bar (i-1,j0-1,i+1,j0+1);
      x:=(i-i0)/E;
      Str(x:0:0,S);
      If x<0 then OutTextXY(i+2,j0+5,S);
    end;
For j:=51 to 399 do
  If ((j-j0) mod E=0) then
    begin
      y:=(j-j0)/E;
      Bar (i0-1,j-1,i0+1,j+1);
      Str(y:0:0,S);
      If y<0 then OutTextXY(i0+5,j,S);
    end;
x:=1; {tiem can dung}
i:=round(x*E+i0);
Line(i,50,i,400);
{tiem can xien}
i:=170; y:=(i-i0)/E; {y=x}
j:=round(-y*E+j0);
il:=480; y:=(il-i0)/E; {y=x}
jl:=round(-y*E+j0);
Line(i,j,il,jl);
For i:=70 to 580 do
  begin
    x:=(i-i0)/E;
    If (x-1)<0 then

```

```

begin
    y:=f(x);
    j:=round(-y*e+j0);
    SetColor(2);
    If (i<70)and(j>70)and(j<380) then Line(i,j,il,jl);
    il:=i; jl:=j;
end;
end;
Readln;
CloseGraph;
END.

```

11. Program P20811; {ve do thi ham $x = at + x_0$, $y = bt + y_0$ }

Uses Crt, Graph;

Const i0=300; j0=200; E=80;

x0=1; y0=-1; a=2; b=1;

Var Gd, Gm, i, j, il, jl: Integer;

x, yy: real;

S: string;

Function y(tt: real): real; {tinh y theo t}

Begin

y:=b*tt+y0;

End;

Function t(x: real): real; {tinh t theo x}

Begin

t:=(x-x0)/a;

End;

BEGIN

Gd:=Detect;

InitGraph(Gd, Gm, '');

Line(i0,50,i0,400); {Oy}

Line(50,j0,600,j0); {Ox}

```

OutTextXY(i0+5,j0+5,'0');
OutTextXY(600-3,j0+5,'x');
OutTextXY(i0+5,50,'y');
For i:=51 to 599 do
  If ((i-i0) mod E=0) then
    begin
      Bar (i-1,j0-1,i+1,j0+1);
      x:=(i-i0)/E;
      Str(x:0:0,S);
      If x<>0 then OutTextXY(i+2,j0+5,S);
    end;
For j:=51 to 399 do
  If ((j-j0) mod E=0) then
    begin
      yy:=(j-j0)/E;
      Bar (i0-1,j-1,i0+1,j+1);
      Str(yy:0:0,S);
      If yy<>0 then OutTextXY(i0+5,j,S);
    end;
For i:=70 to 580 do
  begin
    x:=(i-i0)/E;
    j:=round(-y(t(x))*e+j0);
    SetColor(2);
    If (i<>70)and(j>70)and(j<380) then Line(i,j,il,jl);
    il:=i; jl:=j;
  end;
Readln;
CloseGraph;
END.

```

13. Program P20813; {ve do thi ham $x = r\cos t + a$, $y = r\sin t + b$ }

Uses Crt,Graph;

Const i0=300; j0=200; E=60;

x0=1; y0=0; a=3; b=2;

```

Var Gd,Gm,i,j,t: Integer;
    x,y: real;
    S: string;

```

```

Function yy(t: real): real; {tinh y theo t}
Begin
    yy:=b*sin(t)+y0;
End;

```

```

Function xx(t: real): real; {tinh x theo t}
Begin
    xx:=a*cos(t)+x0;
End;

```

```

BEGIN
Gd:=Detect;
InitGraph(Gd, Gm, ' ');
Line(i0,50,i0,400); {Oy}
Line(50,j0,600,j0); {Ox}
OutTextXY(i0+5,j0+5,'0');
OutTextXY(600-3,j0+5,'x');
OutTextXY(i0+5,50,'y');
For i:=51 to 599 do
    If ((i-i0) mod E=0) then
        begin
            Bar (i-1,j0-1,i+1,j0+1);
            x:=(i-i0)/E;
            Str(x:0:0,S);
            If x<>0 then OutTextXY(i+2,j0+5,S);
        end;
For j:=51 to 399 do
    If ((j-j0) mod E=0) then
        begin
            y:=-(j-j0)/E;

```

Bar (i0-1,j-1,i0+1,j+1);
 Str(y:0:0,S);
 If $y < 0$ then OutTextXY(i0+5,j,S);
 end;
 For t:=0 to 1000 do
 begin
 i:=round(xx(t)*E+i0);
 j:=round(-yy(t)*E+j0);
 SetColor(2);
 If (i>70)and(i<600)and(j>70)and(j<380) then
 PutPixel(i,j,2);
 end;
 Readln;
 CloseGraph;
 END.

15. Tọa độ C và D tính như sau:

$$C = (x_2, y_1)$$

$$D = (x_1, y_2)$$

17. Program P20817; {Duong cong Pascal}

Uses Crt,Graph;

Const i0=200; j0=240; E=80;

a=2; b=1;

Var Gd,Gm,i,j,tt: Integer;

x,y,t: real;

S: string;

Function yy(t: real): real; {tinh y theo t}

Begin

yy:=a*sin(t)*cos(t)+b*sin(t);

End;

Function xx(t: real): real; {tinh x theo t}

Begin

xx:=a*sqr(cos(t))+b*cos(t);

```

End;

BEGIN
Gd:=Detect;
InitGraph(Gd, Gm, '');
Line(i0,50,i0,400); {Oy}
Line(50,j0,600,j0); {Ox}
OutTextXY(i0+5,j0+5,'0');
OutTextXY(600-3,j0+5,'x');
OutTextXY(i0+5,50,'y');
For i:=51 to 599 do
  If ((i-i0) mod E=0) then
    begin
      Bar (i-1,j0-1,i+1,j0+1);
      x:=(i-i0)/E;
      Str(x:0:0,S);
      If x<>0 then OutTextXY(i+2,j0+5,S);
    end;
For j:=51 to 399 do
  If ((j-j0) mod E=0) then
    begin
      y:=(j-j0)/E;
      Bar (i0-1,j-1,i0+1,j+1);
      Str(y:0:0,S);
      If y<>0 then OutTextXY(i0+5,j,S);
    end;
For tt:=0 to 6300 do {tt:0..2pi}
  begin
    t:=tt/1000;
    x:=xx(t);
    y:=yy(t);
    i:=Round(x*e+i0);
    j:=Round(y*e+j0);
    If (i>70)and(j>70)and(i<600)and(j<400) then

```

PutPixel(i,j,2);
 end;
 Readln;
 CloseGraph;
 END.

19. Program P20819; {Duong cong Astroid}

Uses Crt,Graph;
 Const i0=300; j0=240; E=70;
 b=2;
 Var Gd,Gm,i,j,tt: Integer;
 x,y,t: real;
 S: string;

Function xx(t: real): real; {tinh x theo t}
 Begin
 xx:=b*cos(t)*Sqr(cos(t));
 End;

Function yy(t: real): real; {tinh y theo t}
 Begin
 yy:=b*sin(t)*Sqr(sin(t));
 End;

BEGIN
 Gd:=Detect;
 InitGraph(Gd, Gm, ' ');
 Line(i0,50,i0,400); {Oy}
 Line(50,j0,600,j0); {Ox}
 OutTextXY(i0+5,j0+5,'0');
 OutTextXY(600-3,j0+5,'x');
 OutTextXY(i0+5,50,'y');
 For i:=51 to 599 do
 If ((i-i0) mod E=0) then

```

begin
    Bar (i-1,j0-1,i+1,j0+1);
    x:=(i-i0)/E;
    Str(x:0:0,S);
    If x<>0 then OutTextXY(i+2,j0+5,S);
end;
For j:=51 to 399 do
    If ((j-j0) mod E=0) then
        begin
            y:=(j-j0)/E;
            Bar (i0-1,j-1,i0+1,j+1);
            Str(y:0:0,S);
            If y<>0 then OutTextXY(i0+5,j,S);
        end;
For tt:=0 to 6300 do {tt:0..2pi}
    begin
        t:=tt/1000;
        x:=xx(t);
        y:=yy(t);
        i:=Round(x*e+i0);
        j:=Round(y*e+j0);
        If (i>70)and(j>70)and(i<600)and(j<400) then
            PutPixel(i,j,2);
        end;
Readln;
CloseGraph;
END.

```

21. Program P20821; {Duong cong Strofoid}

```

Uses Crt,Graph;
Const i0=300; j0=240; E=70;
      a=2;
Var Gd,Gm,i,j,tt: Integer;
    x,y,t: real;

```


S: string;

Function xx(t: real): real; {tính x theo t}

Begin

xx:=a*(Sqr(t)-1)/(1+Sqr(t));

End;

Function yy(t: real): real; {tính y theo t}

Begin

yy:=a*t*(Sqr(t)-1)/(1+Sqr(t));

End;

BEGIN

Gd:=Detect;

InitGraph(Gd, Gm, '');

Line(i0,50,i0,400); {Oy}

Line(50,j0,600,j0); {Ox}

OutTextXY(i0+5,j0+5,'0');

OutTextXY(600-3,j0+5,'x');

OutTextXY(i0+5,50,'y');

For i:=51 to 599 do

 If ((i-i0) mod E=0) then

 begin

 Bar (i-1,j0-1,i+1,j0+1);

 x:=(i-i0)/E;

 Str(x:0:0,S);

 If x<>0 then OutTextXY(i+2,j0+5,S);

 end;

For j:=51 to 399 do

 If ((j-j0) mod E=0) then

 begin

 y:=(j-j0)/E;

 Bar (i0-1,j-1,i0+1,j+1);

 Str(y:0:0,S);

.xh.
628.6

```

        If  $y < 0$  then OutTextXY(i0+5,j,S);
    end;
For tt:=-3000 to 3000 do
    begin
        t:=tt/1000;
        x:=xx(t);
        y:=yy(t);
        i:=Round(x*e+i0);
        j:=Round(y*e+j0);
        If (i>70)and(j>70)and(i<600)and(j<400) then
PutPixel(i,j,2);
        end;
    Readln;
    CloseGraph;
END.

```

Bài 9. Các thuật toán tạo chuyển động trên đồ họa

1. Program P20901; {hình vuông chuyển động đều từ trái sang phải }

Uses Crt, Graph;

Var Gd,Gm,x,y: Integer;

Pa,Pb: Pointer;

Size: Word;

BEGIN

Gd:=Detect;

InitGraph(Gd,Gm,' ');

Size:=ImageSize(0,0,10,10);

GetMem(Pb,Size);

GetImage(0,0,10,10,Pb^);

GetMem(Pa,Size);

Bar(0,0,10,10);

GetImage(0,0,10,10,Pa^);

ClearDevice;

x:=0; y:=200;

Repeat

PutImage(x,y,Pa^,NormalPut);

Delay(10);

PutImage(x,y,Pb^,NormalPut);

Inc(x);

If x>600 then x:=0;

Until KeyPressed;

CloseGraph;

END.

3. Program P20903; {điều khiển vận tốc của hình vuông}

Uses Crt, Graph;

Var Gd,Gm,x,y,v: Integer;

Pa,Pb: Pointer;

Size: Word;
c: char;

```
BEGIN
Gd:=Detect;
InitGraph(Gd,Gm,' ');
Size:=ImageSize(0,0,10,10);
GetMem(Pb,Size);
GetImage(0,0,10,10,Pb^);
GetMem(Pa,Size);
Bar(0,0,10,10);
GetImage(0,0,10,10,Pa^);
ClearDevice;
x:=0; y:=200; v:=20;
Repeat
  Repeat
    PutImage(x,y,Pa^,NormalPut);
    Delay(v);
    PutImage(x,y,Pb^,NormalPut);
    Inc(x);
    If x>600 then x:=0;
  Until KeyPressed;
  c:=ReadKey; If c= #0 then c:=ReadKey;
  If (c= #72)and(v>0) then Dec(v);
  If (c= #80)and(v<100) then Inc(v);
Until (c= #27)or(c= #13);
CloseGraph;
END.
```

5. Program P20905; {hình vuông chuyển động xiên đều}
Uses Crt, Graph;
Const a=-pi/4; d=5; v=200;
Var Gd,Gm,x,y,dx,dy: Integer;
Pa,Pb: Pointer;
Size: Word;

```

BEGIN
  Gd:=Detect;
  InitGraph(Gd,Gm,' ');
  Size:=ImageSize(0,0,10,10);
  GetMem(Pb,Size);
  GetImage(0,0,10,10,Pb^);
  GetMem(Pa,Size);
  Bar(0,0,10,10);
  GetImage(0,0,10,10,Pa^);
  ClearDevice;
  x:=300; y:=200; dx:=round(d*cos(a)); dy:=round(d*sin(a));
  Repeat
    PutImage(x,y,Pa^,NormalPut);
    Delay(v);
    PutImage(x,y,Pb^,NormalPut);
    Inc(x,dx); Inc(y,dy);
    If x>600 then x:=0;
    If x<0 then x:=600;
    If y>440 then y:=0;
    If y<0 then y:=440;
  Until KeyPressed;
  CloseGraph;
END.

```

7. Program P20907; {diem chuyen dong tron co van toc}

Uses Crt, Graph;

Const r=150;

Var Gd,Gm,x0,y0,x,y,v: Integer;

a: real; {goc}

c: char;

BEGIN

Gd:=Detect;

InitGraph(Gd,Gm,' ');

x0:=GetMaxX div 2;

```

y0:=GetMaxY div 2;
PutPixel(x0,y0,4);
v:=10; a:=0;
Repeat
  Repeat
    x:=x0+Round(r*cos(a)); y:=y0+Round(r*sin(a));
    PutPixel(x,y,15);
    Delay(v);
    PutPixel(x,y,0);
    a:=a+0.01;
  Until KeyPressed;
  c:=ReadKey; If c= #0 then c:=ReadKey;
  If (c= #72)and(v>0) then Dec(v);
  If (c= #80)and(v<100) then Inc(v);
Until (c= #13)or(c= #27);
CloseGraph;
END.

```

9. Program P20909; {doan thang tu quay}

```

Uses Crt, Graph;
Const l=150; v=5; x0=200; y0=200; t=0.01;
Var Gd,Gm,x,y: Integer;
    a: real; {goc}
BEGIN
  Gd:=Detect;
  InitGraph(Gd,Gm,' ');
  a:=0;
  Repeat
    x:=x0+Round(l*cos(a));
    y:=y0+Round(l*sin(a));
    SetColor(15);
    Line(x0,y0,x,y);
    Delay(v);
    SetColor(0);

```

```

    Line(x0,y0,x,y);
    a:=a+t;
Until KeyPressed;
CloseGraph;
END.

```

11. Program P20911; {doan thang tu quay}

```

Uses Crt, Graph;
Const l1=100; l2=150; v=5; x0=200; y0=200; t=0.01; b=-0.9;
Var Gd,Gm,x1,y1,x2,y2: Integer;
    a: real; {goc}
BEGIN
Gd:=Detect;
InitGraph(Gd,Gm,' ');
a:=0;
Repeat
    Bar(x0-1,y0-1,x0+1,y0+1);
    x1:=x0+Round(l1*cos(a));
    y1:=y0+Round(l1*sin(a));
    x2:=x0+Round(l2*cos(a+b));
    y2:=y0+Round(l2*sin(a+b));
    SetColor(15);
    Line(x1,y1,x2,y2);
    Delay(v);
    SetColor(0);
    Line(x1,y1,x2,y2);
    a:=a+t;
Until KeyPressed;
CloseGraph;
END.

```

13. Program P20913; {hình vuông đặc tu quay}

```

Uses Crt, Graph;
Const w = 150; v = 0;

```

```

        x0 = 200; y0 = 200;
        t = 0.05; b = pi/2;
Var   Gd,Gm,x1,y1,x2,y2,x3,y3,x4,y4,l: Integer;
        a: real;      {goc}
BEGIN
    Gd:=Detect;
    InitGraph(Gd,Gm,' ');
    SetColor(4);
    a:=0; l:=Round(w/Sqrt(2));
    Repeat
        x1:=x0+Round(l*cos(a));
        y1:=y0+Round(l*sin(a));
        x2:=x0+Round(l*cos(a+b));
        y2:=y0+Round(l*sin(a+b));
        x3:=x0+Round(l*cos(a+2*b));
        y3:=y0+Round(l*sin(a+2*b));
        x4:=x0+Round(l*cos(a+3*b));
        y4:=y0+Round(l*sin(a+3*b));
        Line(x1,y1,x2,y2); Line(x2,y2,x3,y3);
        Line(x3,y3,x4,y4); Line(x1,y1,x4,y4);
        SetFillStyle(1,4); FloodFill(x0,y0,4);
        Delay(v);
        SetFillStyle(1,0); FloodFill(x0,y0,0);
        a:=a+t;
    Until KeyPressed;
    CloseGraph;
END.

```

15. Program P20915; {hình vuông đặc tu quay + chuyển động}

```

Uses Crt, Graph;
Const  w = 150; v = 0; t = 0.01; b = pi/2;
Var   Gd,Gm,x0,y0,x1,y1,x2,y2,x3,y3,x4,y4,l: Integer;
        a: real;      {goc}
BEGIN

```



```

Gd:=Detect;
InitGraph(Gd,Gm,"");
SetColor(4);
a:=0; l:=Round(w/Sqrt(2));
x0:=0; y0:=240;
Repeat
  x1:=x0+Round(l*cos(a));
  y1:=y0+Round(l*sin(a));
  x2:=x0+Round(l*cos(a+b));
  y2:=y0+Round(l*sin(a+b));
  x3:=x0+Round(l*cos(a+2*b));
  y3:=y0+Round(l*sin(a+2*b));
  x4:=x0+Round(l*cos(a+3*b));
  y4:=y0+Round(l*sin(a+3*b));
  Line(x1,y1,x2,y2); Line(x2,y2,x3,y3);
  Line(x3,y3,x4,y4); Line(x1,y1,x4,y4);
  SetFillStyle(1,4); FloodFill(x0,y0,4);
  Delay(v);
  SetFillStyle(1,0); FloodFill(x0,y0,0);
  a:=a+t;
  x0:=x0+1;
  If x0=640 then x0:=0;
Until KeyPressed;
CloseGraph;
END.

```

Bài 10. Các bài toán hình học

1. Program P21001; {4 hình chu nhật}

Uses Crt;

Var c:Array[1..4,1..2] of byte;
 d:Array[1..200,1..200] of byte;
 dx:Array[1..4] of byte;
 i:byte;
 s,n:integer;

Procedure put(a,b,x,y,t:byte);

Var i,j:byte;

Begin

For i:=y to y+b-1 do

 For j:=x to x+a-1 do d[i,j]:=t;

End;

Function test(a,b,x,y:byte):boolean;

Var i,j:byte;

Begin

test:=false;

If y+b-1 > n then exit;

If x+a-1 > n then exit;

For i:=y to y+b-1 do

 For j:=x to x+a-1 do

 If d[i,j]=1 then exit;

test:=true;

End;

Procedure tim;

Var i,a,b,x,y:byte;

 ok:boolean;

Begin

Ok:=false;

```

For y:=1 to n do
  For x:=1 to n do
    If d[y,x]=0 then
      Begin
        ok:=true;
        For i:=1 to 4 do
          If dx[i]=0 then
            Begin
              dx[i]:=1;
              a:=c[i,1];
              b:=c[i,2];
              If test(a,b,x,y) then
                Begin
                  put(a,b,x,y,1);
                  tim;
                  put(a,b,x,y,0);
                End;
              a:=c[i,2];
              b:=c[i,1];
              If test(a,b,x,y) then
                Begin
                  put(a,b,x,y,1);
                  tim;
                  put(a,b,x,y,0);
                End;
              dx[i]:=0;
            End;
          End;
        exit;
      End;
    If ok=false then
      Begin
        Write(' Xep duoc thanh hinh vuong');
        readln;
        halt;
      End;
    End;
  End;
End;

```

```

    End;
End;

BEGIN
Clrscr;
s:=0;
For i:=1 to 4 do
    begin
        Write(' Nhap toa do hinh chu nhath thu ',i,
');readln(c[i,1],c[i,2]);
        s:=s+c[i,1]*c[i,2];
    end;
If round(sqrt(s))<>sqrt(s) then
    begin
        Write('Khong the xep duoc thanh hinh vuong');
        readln;
        halt;
    end;
n:=round(sqrt(s));
fillchar(dx,sizeof(dx),0);
Clrscr;
tim;
END.

```

3. Program P21003;

{Kiem tra 3 so co tao thanh tam giac gi}

Uses Crt;

Var a,b,c: real;

Function Vuong(c1,c2,c3:real): boolean;

Begin

Vuong:=c1*c1=c2*c2+c3*c3;

End;

Function tu(c1,c2,c3:real): boolean;

Begin

tu:=c1*c1>c2*c2+c3*c3;

End;

BEGIN

Repeat

Clrscr;

Write('Nhap do dai 3 canh cua 1 tam giac: ');

Readln(a,b,c);

Until (a>0)and(b>0)and(c>0)and(a<b+c)and(b<c+a)and(c<a+b);

If Vuong(a,b,c)or Vuong(b,c,a)or Vuong(c,b,a) then Write('Tam giac do la tam giac vuong.')

Else If tu(a,b,c)or tu(b,c,a)or tu(c,a,b) then Write('Tam giac do la tam giac tu.')

Else Write('Tam giac do la tam giac nhon.');

Readln;

END.

5. Program P21005; {Tim dinh con lai cua hinh chu nhac}

Uses Crt;

Var i: byte;

x,y: array [1..4] of real;

Procedure Error;

Begin

Write('Du lieu sai');

Readln;

halt;

End;

BEGIN

Clrscr;

For i:=1 to 3 do

begin

Write('Nhap toa do dinh thu ',i,' (x,y) = ');

```

        Readln(x[i],y[i]);
    end;
    If (x[1]=x[2]) and (x[2]=x[3]) then Error;
    If (y[1]=y[2]) and (y[2]=y[3]) then Error;
    If (x[1]=x[2]) then x[4]:=x[3]
        Else If (x[2]=x[3]) then x[4]:=x[1]
            Else If (x[3]=x[1]) then x[4]:=x[2]
                Else Error;
    If (y[1]=y[2]) then y[4]:=y[3]
        Else If (y[2]=y[3]) then y[4]:=y[1]
            Else If (y[3]=y[1]) then y[4]:=y[2]
                Else Error;
    Clrscr;
    Write('Toa do dinh thu 4 la (x,y) = ',x[4]:5:2,',',y[4]:5:2);
    Readln;
END.

```

7. Program P21007;

{Tinh tong khoang cach tu 1 diem toi cac canh cua hinh chu nhat}

Uses Crt;

Var x,y,kc1,kc2:real;

Procedure Tinh(x1,y1,x2,y2: real);

Begin

If (x1<=x)and(x<=x2) then kc1:=x2-x1

Else If x<x1 then kc1:=x2-x1+2*(x1-x)

Else kc1:=x2-x1+2*(x-x2);

If (y1<=y)and(y<=y2) then kc2:=y2-y1

Else If y<y1 then kc2:=y2-y1+2*(y1-y)

Else kc2:=y2-y1+2*(y-y2);

End;

BEGIN

Clrscr;

Write('Nhap toa do diem A: '); Readln(x,y);

```

Clrscr;
Tinh(-0.5,-0.5,0.5,0.5);
Writeln('a. Tong khoang cach la: ',kc1+kc2:5:2);
Tinh(0,0,1,1);
Writeln('b. Tong khoang cach la: ',kc1+kc2:5:2);
Readln;
END.

```

9. Program P21009; {Diem tron n duong tron}.

Uses crt;

Const fi = 'P21009.inp';

Var f:text;

c:array[1..100,1..3] of real;

d,x,y,min:real;

n,i,t:byte;

function inside(x,y:real):boolean;

Var i:byte;

Begin

inside:=false;

For i:=1 to n do

If $\text{sqr}(x-c[i,2]) + \text{sqr}(y-c[i,1]) > \text{sqr}(c[i,3])$ then exit;

inside:=true;

End;

BEGIN

Clrscr;

Assign(f,fi);

Reset(f);

Readln(f,n);

min:=10000;

For i:=1 to n do

Begin

Readln(f,c[i,1],c[i,2],c[i,3]);

If $c[i,3] < \text{min}$ then

Begin

```

        min:=c[i,3];
        t:=i;
    End;
End;
y:=c[t,1]-c[t,3];
d:=0.1;
Repeat
    x:=c[t,2]-c[t,3];
    Repeat
        x:=x+d;
        If inside(x,y) then
            Begin
                Writeln('Diem (',x:3:2,',',y:3:2,') nam trong tat ca cac
hinh tron');
                Readln;
                Exit;
            End;
        Until x>c[t,2]+c[t,3];
        y:=y+d;
        Until y>c[t,1]+c[t,3];
    Close(f);
END.

```

11. Program P21011; {Tim tam giac cuc tieu}

Uses Crt;

Const max = 100;

e = 0.000001;

Var i,n,d1,d2,d3,d4: word;

x,y: array [1..max] of real;

c1,c2,c3,p,dth: real;

Function kc(cs1,cs2: word): real;

Begin

kc:=sqrt(sqrt(x[cs1]-x[cs2])+sqrt(y[cs1]-y[cs2]));

End;


```
Function dt(d1,d2,d3: word): real;
Begin
  c1:=kc(d1,d2);
  c2:=kc(d2,d3);
  c3:=kc(d3,d1);
  p:=(c1+c2+c3)/2;
  dt:=sqrt(p*(p-c1)*(p-c2)*(p-c3));
End;
```

```
Function Found: boolean;
Var i: word;
Begin
  For i:=1 to n do
    If (i<>d1)and(i<>d2)and(i<>d3)and (abs(dth-dt(d1,d2,i)-
dt(d2,d3,i)-dt(d3,d1,i))<e) then
      begin
        d4:=i;
        Found:=true;
        exit;
      end;
  Found:=false;
End;
```

```
BEGIN
  Repeat
    Clrscr;
    Write('Co bao nhieu diem: ');
    Readln(n);
  Until n>2;
  For i:=1 to n do
    begin
      Write('Nhap toa do diem thu ',i,' ');
      Readln(x[i],y[i]);
```

```

    end;
    d1:=1;
    d2:=2;
    d3:=3;
    dth:=dt(d1,d2,d3);
    While Found do
        begin
            d3:=d4;
            dth:=dt(d1,d2,d3);
        end;
    Clrscr;
    Write('Tam giac cuc tieu co 3 dinh la: ',d1:5,d2:5,d3:5);
    Readln;
END.

```

13. Program P21013; {dien tích hợp của 3 đường tròn}

```

Uses crt;
Const fi = 'P21013.inp';
Var a,b,c,oa,ob,oc,s,ga,gb,gc:real;
    f:text;
BEGIN
    Clrscr;
    Assign(f,fi);
    Reset(f);
    Readln(f,oa,ob,oc);
    c:=sqrt(oa*oa+ob*ob+oa*ob);
    b:=sqrt(oa*oa+oc*oc+oa*oc);
    a:=sqrt(oc*oc+ob*ob+oc*ob);
    ga:=(b*b+c*c-a*a)/(2*b*c);
    gb:=(a*a+c*c-b*b)/(2*a*c);
    gc:=(b*b+a*a-c*c)/(2*a*b);
    ga:= 2*ArcTan (sqrt (1-sqr (ga)) /ga);
    gb:= 2*ArcTan (sqrt (1-sqr (gb)) /gb);
    gc:= 2*ArcTan (sqrt (1-sqr (gc)) /gc);
    s:=(oa*ob+ob*ob+oc*oa)*sqrt(3)/2;

```

```

s:=s+(2*pi-ga)*oa*oa/2+(2*pi-gb)*ob*ob/2+(2*pi-gc)*oc*oc/2;
Writeln('dien tich phan hop cua 3 duong tron la: ',s);
Close(f);

```

```

Readln;
END.

```

15. Program P21015;{Dien tich cua hop n hinh chu nhât}

```

Uses crt;

```

```

Const fi = 'P21016.inp';

```

```

Type mang=Array[1..101] of real;

```

```

Var n:byte;

```

```

    x,y:mang;

```

```

    dx,dy:byte;

```

```

    d:array[1..200,1..4] of real;

```

```

    c:array[1..200,1..200] of byte;

```

```

    f:text;

```

```

Procedure cut(t:byte;Var len:byte;Var m:mang);

```

```

Var i:byte;

```

```

Begin

```

```

    Dec(len);

```

```

    For i:=t to len do m[i]:=m[i+1];

```

```

End;

```

```

Procedure xapxep(Var len:byte;Var m:mang);

```

```

Var i,j:byte;

```

```

d:real;

```

```

Begin

```

```

    For i:=1 to len-1 do

```

```

        For j:=i+1 to len do

```

```

            If m[j]<m[i] then

```

```

                Begin

```

```

                    d:=m[j];

```

```

                    m[j]:=m[i];

```

```

                    m[i]:=d;

```

```

        End;
    i:=1;
    While (i<len) do If m[i]=m[i+1] then cut(i,len,m) else inc(i);
End;

```

```

Procedure input;
Var i,j:byte;
Begin
    dx:=0;
    dy:=0;
    Assign(f,fi);
    Reset(f);
    Readln(f,n);
    For i:=1 to n do
        Begin
            Readln(f,d[i,1],d[i,2],d[i,3],d[i,4]);
            Inc(dx,2);
            Inc(dy,2);
            x[dx-1]:=d[i,2];
            y[dy-1]:=d[i,1];
            x[dx]:=d[i,4];
            y[dy]:=d[i,3];
        End;
    Close(f);
    xapxep(dx,x);
    xapxep(dy,y);
End;

```

```

Procedure work;
Var i,j,k,x1,y1,x2,y2:byte;
    sum:real;
Begin
    Fillchar(c,sizeof(c),0);
    For k:=1 to n do

```

```

Begin
  For x1:=1 to dx do If d[k,2]=x[x1] then break;
  For x2:=1 to dx do If d[k,4]=x[x2] then break;
  For y1:=1 to dy do If d[k,1]=y[y1] then break;
  For y2:=1 to dy do If d[k,3]=y[y2] then break;
  For i:=y1 to y2-1 do
    For j:=x1 to x2-1 do c[i,j]:=1;
  End;
sum:=0;
For i:=1 to dy-1 do
  For j:=1 to dx-1 do sum:=sum+c[i,j]*(y[i+1]-y[i])*(x[j+1]-
x[j]);
  Writeln;
  Write(' Tong dien tich che phu la: ',sum:6:3);
End;

```

BEGIN

Clrscr;

input;

work;

Readln;

END.

17. Program P21017; {Tim bao dong loi}

Uses Crt;

Const max =100;

Var i,j,n,l: word;

tam: integer;

x,y: array [1..max] of real;

d: array [1..max] of word;

cx: array [1..max] of boolean;

a,b,c: real;

found,stop: boolean;

```

Procedure Gan(am,bm,cm: real);
Begin
  a:=am;
  b:=bm;
  c:=cm;
End;

```

```

Procedure Timdt(cs1,cs2: word);
Begin
  If x[cs1]=x[cs2] then
    begin
      Gan(-1,0,x[cs1]);
      exit;
    end;
  If y[cs1]=y[cs2] then
    begin
      Gan(0,-1,y[cs1]);
      exit;
    end;
  a:=1;
  b:=(x[cs1]-x[cs2])/(y[cs2]-y[cs1]);
  c:=-x[cs1]-b*y[cs1];
End;

```

```

Function f(cs: word): real;
Begin
  f:=a*x[cs]+b*y[cs]+c;
End;

```

```

Function Dung(cs1,cs2: word): boolean;
Var i: word;
Begin
  Timdt(cs1,cs2);
  Tam:=0;

```

```

For i:=1 to n do
  If (i > cs1) and (i > cs2) then
    begin
      If f(i) < 0 then tam:=-1 Else tam:=1;
      break;
    end;
For i:=1 to n do
  If (i > cs1) and (i > cs2) then
    If f(i)*tam < 0 then
      begin
        dung:=false;
        exit;
      end;
  dung:=true;
End;

```

```

BEGIN
  Clrscr;
  Write('Co bao nhieu diem: ');
  Readln(n);
  For i:=1 to n do
    begin
      Write('Nhap toa do diem thu ',i,' ');
      Readln(x[i],y[i]);
    end;
  Found:=false;
  FillChar(cx,SizeOf(cx),true);
  For i:=1 to n-1 do
    begin
      For j:=i+1 to n do
        If dung(i,j) then
          begin
            l:=2;
            d[1]:=i;

```

```

        d[2]:=j;
        cx[i]:=false;
        cx[j]:=false;
        found:=true;
        break;
    end;
    If found then break;
end;
stop:=false;
While not stop do
    begin
        stop:=true;
        For i:=1 to n do
            If cx[i] and dung(d[1],i) then
                begin
                    stop:=false;
                    Inc(l);
                    d[1]:=i;
                    cx[i]:=false;
                end;
            end;
        end;
        Clrscr;
        Writeln('Da giai loi chua tat ca co ',l,' dinh la cac dinh: ');
        For i:=1 to l do
            begin
                Write(d[i]:5);
                If i mod 10=0 then Writeln;
            end;
        Readln;
    END.

```

19. Program P21019; {Kiem tra da giai}
 Uses Crt;
 Const max =100;
 Var i,j,n: word;


```

tam: integer;
x,y: array [1..max+1] of real;
a,b,c: real;
ok1,ok2: boolean;

```

```

Procedure Gan(am,bm,cm: real);
Begin
  a:=am;
  b:=bm;
  c:=cm;
End;

```

```

Procedure Timdt(cs1,cs2: word);
Begin
  If x[cs1]=x[cs2] then
    begin
      gan(-1,0,x[cs1]);
      exit;
    end;
  If y[cs1]=y[cs2] then
    begin
      gan(0,-1,y[cs1]);
      exit;
    end;
  a:=1;
  b:=(x[cs1]-x[cs2])/(y[cs2]-y[cs1]);
  c:=-x[cs1]-b*y[cs1];
End;

```

```

Function f(cs: word): real;
Begin
  f:=a*x[cs]+b*y[cs]+c;
End;

```

```

Function Cat(cs1,cs2: word): boolean;
Begin
  Timdt(cs1,cs1+1);
  If f(cs2)*f(cs2+1)<0 then
    begin
      Timdt(cs2,cs2+1);
      If f(cs1)*f(cs1+1)<0 then
        begin
          cat:=true;
          exit;
        end;
      end;
    end;
  Cat:=false;
End;

```

```

Function Dung(cs1,cs2: word): boolean;
Var i: word;
Begin
  Timdt(cs1,cs2);
  Tam:=0;
  For i:=1 to n do
    If (i<>cs1)and(i<>cs2) then
      begin
        If f(i)<0 then tam:=-1 Else tam:=1;
        break;
      end;
  For i:=1 to n do
    If (i<>cs1)and(i<>cs2) then
      If f(i)*tam<0 then
        begin
          Dung:=false;
          exit;
        end;
  Dung:=true;
End;

```

```

BEGIN
  Repeat
    Clrscr;
    Write('Co bao nhieu diem: ');
    Readln(n);
  Until n>2;
  For i:=1 to n do
    begin
      Write('Nhap toa do diem thu 'i,': ');
      Readln(x[i],y[i]);
    end;
  x[n+1]:=x[1];
  y[n+1]:=y[1];
  ok1:=true;
  For i:=1 to n-1 do
    begin
      For j:=i+1 to n do
        If cat(i,j) then
          begin
            ok1:=false;
            break;
          end;
      If not ok1 then break;
    end;
  If not ok1 then ok2:=false
  Else begin
    ok2:=true;
    For i:=1 to n-1 do
      If not Dung(i,i+1) then
        begin
          ok2:=false;
          break;
        end;
  end;

```

609.

```

        If not dung(1,n) then ok2:=false;
    end;
    Clrscr;
    If ok1 then Writeln('Da giac khong tu cat.')
    Else Writeln('Da giac co tu cat.');
```

 If ok2 then Writeln('Da giac la da giac loi.')

 Else Writeln('Da giac khong phai la da giac loi.');

 Readln;

END.

21. Program P21021;{phu roi nhau cua hinh chu nhac CN0}

Uses crt;

Const fi = 'P21021.inp';

Var c:Array[1..100,1..2] of byte;

 d:Array[1..200,1..200] of byte;

 dx:Array[1..4] of byte;

 s,n,i,cx,cy:integer;

 f:text;

Procedure put(a,b,x,y,t:byte);

Var i,j:byte;

Begin

For i:=y to y+b-1 do

 For j:=x to x+a-1 do d[i,j]:=t;

End;

Function test(a,b,x,y:byte):boolean;

Var i,j:byte;

Begin

test:=false;

If y+b-1 > cy then exit;

If x+a-1 > cx then exit;

For i:=y to y+b-1 do

 For j:=x to x+a-1 do If d[i,j]=1 then exit;

test:=true;

For j:=x to x+a-1 do If d[i,j]=1 then exit;
 test:=true;
 End;

Procedure tim(k:byte);

Var i,a,b,x,y:byte;

Begin

If k=n then

Begin

write(' Xep duoc');

readln;

halt;

End;

For y:=1 to cy do

For x:=1 to cx do

If d[y,x]=0 then

Begin

For i:=1 to n do

If dx[i]=0 then

Begin

dx[i]:=1;

a:=c[i,1];

b:=c[i,2];

If test(a,b,x,y) then

Begin

put(a,b,x,y,1);

tim(k+1);

put(a,b,x,y,0);

End;

a:=c[i,2];

b:=c[i,1];

If test(a,b,x,y) then

Begin

put(a,b,x,y,1);

```

                                tim(k+1);
                                put(a,b,x,y,0);
                                End;
                                dx[i]:=0;
                                End;
                                exit;
                                End;
                                End;

BEGIN
  s:=0;
  Assign(f,fi);
  Reset(f);
  Readln(f,n,cx,cy);
  For i:=1 to n do
    Begin
      readln(f,c[i,1],c[i,2]);
      s:=s+c[i,1]*c[i,2];
    End;
  Close(f);
  If s<>cx*cy then
    Begin
      write('Khong the xep duoc thanh hinh vuong');
      readln;
      halt;
    End;
  fillchar(dx,sizeof(dx),0);
  Clrscr;
  tim(0);
END.

```

Bài 11. Ôn tập

1. Program P21101; {So sanh hai File}

Uses Crt;

Const inp1='P21101a.inp';

inp2='P21101b.inp';

Var f1,f2: file of byte;

c1,c2: byte;

BEGIN

Clrscr;

Assign(f1,inp1); Reset(f1);

Assign(f2,inp2); Reset(f2);

While (not Eof(f1))and(not Eof(f2)) do

begin

Read(f1,c1); Read(f2,c2);

If c1 <> c2 then

begin

Writeln('Hai file khác nhau'); Readln;

Close(f1); Close(f2);

exit;

end;

end;

If (not Eof(f1))or(not Eof(f2)) then Writeln('Hai file khác nhau.')

Else Writeln('Hai file giống nhau.');

Close(f1); Close(f2);

Readln;

END.

3. {\$N+}

Program P21103; {Hình vuông chuyển động trên quỹ đạo

Ellipse}

Uses Crt,Graph;

Const GocLechHV = 4*pi/180;

```

    MauHinhVuong=1;
    BanKinhHV=50;
    a=250;
    b=80;
Var  gd,gm: integer;
    xHV,yHV: array [1..4] of integer;
    xTam,yTam,xG,yG: integer;
    GocXoayHV,temp: real;
    DoDoiX: integer;

Procedure KhoiTao;
Begin
    xG:=320;
    yG:=240;
    Randomize;
    GocXoayHV:=random(180)*pi/180;
    xTam:=random(a);
    DoDoiX:=1;
End;

Procedure VeHinhVuong(Mau: byte);
Var i: byte;
Begin
    For i:=1 to 4 do
        begin
            xHV[i]:=xG+xTam+round(BanKinhHV*sin(GocXoayHV+(i-1)*(pi/2)));
            yHV[i]:=yG+yTam+round(BanKinhHV*cos(GocXoayHV+(i-1)*(pi/2)));
        end;
    SetColor(Mau);
    For i:=1 to 3 do line(xHV[i],yHV[i],xHV[i+1],yHV[i+1]);
    line(xHV[1],yHV[1],xHV[4],yHV[4]);
    SetFillStyle(1,Mau);

```



```

        MauHinhVuong=1;
        BanKinhHV=50;
        a=250;
        b=80;
Var   gd,gm: integer;
        xHV,yHV: array [1..4] of integer;
        xTam,yTam,xG,yG: integer;
        GocXoayHV,temp: real;
        DoDoiX: integer;

```

```

Procedure KhoiTao;
Begin
    xG:=320;
    yG:=240;
    Randomize;
    GocXoayHV:=random(180)*pi/180;
    xTam:=random(a);
    DoDoiX:=1;
End;

```

```

Procedure VeHinhVuong(Mau: byte);
Var i: byte;
Begin
    For i:=1 to 4 do
        begin
            xHV[i]:=xG+xTam+round(BanKinhHV*sin(GocXoayHV+(i-1)*(pi/2)));
            yHV[i]:=yG+yTam+round(BanKinhHV*cos(GocXoayHV+(i-1)*(pi/2)));
        end;
    SetColor(Mau);
    For i:=1 to 3 do line(xHV[i],yHV[i],xHV[i+1],yHV[i+1]);
    line(xHV[1],yHV[1],xHV[4],yHV[4]);
    SetFillStyle(1,Mau);

```

```
FloodFill(xG+xTam+3,yG+yTam+3,Mau);
SetColor(15);
Ellipse(xG,yG,0,360,a,b);
Circle(xG+xTam,yG+yTam,2);
End;
```

```
Function Can(x1,a1,b1: real): real;
Var r: real;
Begin
  r:=(1-((x1*x1)/(a1*a1)));
  If r<0.01 then Can:=0
  Else Can:=sqrt(r*b1*b1);
End;
```

```
BEGIN
gd:=detect;
InitGraph(gd,gm,' ');
KhoiTao;
Repeat
  GocXoayHV:=GocXoayHV+GocLechHV;
  If (xTam<-a)or(xTam>a) then DoDoix:=-DoDoix;
  xTam:=xTam+DoDoix;
  If (xTam=-a)or(xTam=a) then temp:=0
  Else temp:=can(xtam,a,b);
  If DoDoix>0 then yTam:=round(temp)
  Else yTam:=-round(temp);
  VeHinhVuong(MauHinhVuong);
  Delay(20);
  VeHinhVuong(0);
Until KeyPressed;
Closegraph;
END.
```

5. Program P21105; {Duong cong Bong hoa}

Uses Crt, Graph;

Const i0=300; j0=240; E=20;

 D=1; B=2; A=7;

Var Gd, Gm, i, j, tt, n: Integer;

 x, y, t, u: real;

 S: string;

Function xx(t: real): real; {tinh x theo t}

Begin

 xx:=(A-B)*cos(t)-D*cos(u);

End;

Function yy(t: real): real; {tinh y theo t}

Begin

 yy:=(A-B)*sin(t)-D*sin(u);

End;

Function USCLN(a, b: integer): integer;

Var r: integer;

Begin

 While b > 0 do begin r:=a mod b; a:=b; b:=r; end;

 USCLN:=a;

End;

BEGIN

 Gd:=Detect;

 InitGraph(Gd, Gm, '');

 Line(i0, 50, i0, 400); {Oy}

 Line(50, j0, 600, j0); {Ox}

 OutTextXY(i0+5, j0+5, '0');

 OutTextXY(600-3, j0+5, 'x');

 OutTextXY(i0+5, 50, 'y');

 For i:=51 to 599 do

 If ((i-i0) mod E=0) then

```

begin
    Bar (i-1,j0-1,i+1,j0+1);
    x:=(i-i0)/E;
    Str(x:0:0,S);
    If x<0 then OutTextXY(i+2,j0+5,S);
end;
For j:=51 to 399 do
    If ((j-j0) mod E=0) then
        begin
            y:=(j-j0)/E;
            Bar (i0-1,j-1,i0+1,j+1);
            Str(y:0:0,S);
            If y<0 then OutTextXY(i0+5,j,S);
        end;
n:=B div USCLN(A,B);
For tt:=0 to n*6300 do {t:0..n*2pi}
    begin
        t:=tt/1000; u:=A*t/B; x:=xx(t); y:=yy(t);
        i:=Round(x*e+i0); j:=Round(y*e+j0);
        If (i>70)and(j>70)and(i<600)and(j<400) then
            PutPixel(i,j,2);
    end;
Readln;
CloseGraph;
END.

```

7. {\$N+}

```

Program P21107; {Doan thang quay quanh diem giua co dinh}
Uses Crt,Graph;
Const GocLech=pi/180;
      MauDoanthang=1;
      DoDai=100;
Var gd,gm: integer;
    xTam,yTam: integer;

```

```

GocXoay: real;

Procedure KhoiTao;
Begin
  Randomize;
  xTam:=random(640);
  yTam:=random(480);
  GocXoay:=random(180)*pi/180;
End;

Procedure VeDoanthang(Mau: byte);
Var x1,x2,y1,y2: integer;
Begin
  x1:=xTam+round(DoDai*cos(GocXoay));
  y1:=yTam+round(DoDai*sin(GocXoay));
  x2:=xTam+round(DoDai*cos(GocXoay+pi));
  y2:=yTam+round(DoDai*sin(GocXoay+pi));
  SetColor(Mau);
  Line(x1,y1,x2,y2);
  SetColor(4);
  Circle(xTam,yTam,2);
End;

BEGIN
  gd:=detect;
  InitGraph(gd,gm,' ');
  KhoiTao;
  Repeat
    GocXoay:=GocXoay+GocLech;
    VeDoanthang(MauDoanthang);
    Delay(30);
    VeDoanthang(0);
  Until keypressed;
  CloseGraph;
END.

```

9. Program P21109; {Cac diem chuyen dong voi van toc khac nhau}

Uses Crt,Graph; {Gap canh man hinh thi phan xa tro lai}

Const BanKinh=3;

Smax=5;

Tre=10;

Nmax=10;

Var n: byte;

gd,gm: integer;

x,y: array [1..Nmax] of integer;

Huong: array [1..Nmax] of real;

MauBong,s: array [1..Nmax] of byte;

Procedure KhoiTao;

Var i: byte;

Begin

Randomize;

n:=random(Nmax-5)+5;

For i:=1 to n do

begin

x[i]:=random(640-2*BanKinh)+BanKinh;

y[i]:=random(480-2*BanKinh)+BanKinh;

MauBong[i]:=random(15)+1;

SetColor(MauBong[i]);

Circle(x[i],y[i],BanKinh);

SetFillStyle(1,MauBong[i]);

FloodFill(x[i],y[i],MauBong[i]);

Huong[i]:=random(360)*pi/180;

s[i]:=random(Smax)+1;

end;

End;

Procedure ChuyenDong;

Var i: byte;

```

Begin
  For i:=1 to n do
    begin
      x[i]:=x[i]+round(s[i]*cos(Huong[i]));
      y[i]:=y[i]+round(s[i]*sin(Huong[i]));
      SetColor(MauBong[i]);
      Circle(x[i],y[i],BanKinh);
      SetFillStyle(1,MauBong[i]);
      FloodFill(x[i],y[i],MauBong[i]);
      If x[i]<=BanKinh then
        If Huong[i]>pi then Huong[i]:=3*pi-Huong[i]
        Else Huong[i]:=pi-Huong[i];
      If x[i]>=640-BanKinh then
        If Huong[i]<pi then Huong[i]:=pi-Huong[i]
        Else Huong[i]:=3*pi-Huong[i];
      If y[i]<=BanKinh then Huong[i]:=2*pi-Huong[i];
      If y[i]>=480-BanKinh then Huong[i]:=2*pi-Huong[i];
    end;
  End;

BEGIN
  gd:=detect;
  InitGraph(gd,gm,' ');
  KhoiTao;
  While not KeyPressed do
    begin
      Delay(Tre);
      ClearDevice;
      ChuyenDong;
    end;
  CloseGraph;
END.

```

```

11. Program P21111;
{Ve cac hinh tron va dem so diem trong moi hinh tron }
Uses Crt,Graph;
Const inp = 'P21107.inp';
        out='P21107.out';
Var    f: text;
        gd,gm: integer;
        m,n: byte;
        a: array [1..100] of byte;
        r: array [1..100] of real;
        x,y: array [1..100] of integer;

```

```

Procedure Nhap;
Var i: byte;
Begin
    Assign(f,inp);Reset(f);
    Readln(f,n);
    For i:=1 to n do Read(f,r[i]);
    Readln(f);
    Readln(f,m);
    For i:=1 to m do Readln(f,x[i],y[i]);
    Close(f);
End;

```

```

Function Can(x1,y1: real): real;
Begin
    Can:=sqrt(sqr(x1)+sqr(y1));
End;

```

```

Function Trong(i,j: byte): boolean;
Begin
    If Can(x[j],y[j])<=r[i] then Trong:=true
    Else Trong:=false;
End;

```



```

Procedure Tinh;
Var i,j: byte;
Begin
  FillChar(a,Sizeof(a),0);
  For i:=1 to n do
    For j:=1 to m do
      If Trong(i,j) then Inc(a[i]);
  Assign(f,out); Rewrite(f);
  For i:=1 to n do Writeln(f,a[i]);
  Close(f);
End;

```

```

Procedure Ve;
Var i:byte;
Begin
  SetColor(green);
  For i:=1 to n do Circle(0,0,round(r[i]));
  SetColor(14);
  For i:=1 to m do Circle(x[i],y[i],2);
End;

```

```

BEGIN
  Nhap;
  Tinh;
  gd:=detect;
  InitGraph(gd,gm,' ');
  Ve;
  Repeat Until KeyPressed;
  CloseGraph;
END.

```

13. Program P21113; {Cap nhat File}
 Uses Crt;

```

Const inp1 = 'P21113a.inp';
      inp2 = 'P21113b.inp';
      out = 'P21113.out';
Type KiemKe = record
    TenHang: string[40];
    TonKho : integer;
end;
HoaDon = record
    TenHang : string[40];
    Nhap,Xuat: integer;
end;
Var f,h: file of KiemKe;
    g: file of HoaDon;
    kk,CapNhat: KiemKe;
    hd: HoaDon;

```

Procedure Nhap;

Begin

```

Assign(f,inp1); Rewrite(f);
Write('Ten hang: '); Readln(Kk.Tenhang) ;
Write('Ton kho :') ; Readln(Kk.Tonkho);
Write(f,kk) ;
Assign(g,inp2); Rewrite(g);
Write('Ten hang: '); Readln(Hd.Tenhang) ;
Write('Nhap : '); Readln(Hd.Nhap);
Write('Xuat : '); Readln(Hd.Xuat);
Write(g,hd);
Readln;
Close(f); Close(g);
end;

```

Procedure ghi;

Begin

```

Assign(f,inp1); Reset(f);

```

```

Assign(g,inp2); Reset(g);
Assign(h,out); Rewrite(h);
While not Eof(f) do
    begin
        Read(f,kk); Read(g,hd);
        CapNhat.TenHang:=kk.TenHang;
        CapNhat.Tonkho:=kk.TonKho+hd.Nhap - hd.Xuat;
        Write(h,CapNhat);
        Write('So luong con lai trong kho la: ',Capnhat.Tonkho)
    end;
Readln;
Close(h); Close(g); Close(f);
End;

```

```

BEGIN
    Nhap;
    Ghi;
END.

```

15. Program P21115; {Tao ma tran lien thong}

```

Uses Crt;
Const nmax =100;
    inp = 'inp.dat';
    out = 'out.dat';
Var f: text;
    n: byte;
    a: array [1..nmax,1..nmax] of byte;
    b: array [1..nmax,1..nmax] of integer;
Procedure Nhap;
Var i,j: byte;
Begin
    Assign(f,inp); Reset(f);
    Readln(f,n);
    For i:=1 to n do

```

```

begin
    For j:=1 to n do Read(f,a[i,j]);
    Readln(f);
end;
Close(f);
End;

Procedure XuLy;
Var i,j,k: byte;
Begin
    For i:=1 to n do
        For j:=1 to n do
            If a[i,j]=1 then b[i,j]:=1 Else b[i,j]:=10000;
        For k:=1 to n do
            For i:=1 to n do
                For j:=1 to n do
                    If b[i,j]>b[i,k]+b[k,j] then b[i,j]:=b[i,k]+b[k,j];
                End;
            End;
        End;
    End;
End;

```

```

Procedure InKQ;
Var i,j: byte;
Begin
    Assign(f,out); Rewrite(f);
    For i:=1 to n do
        begin
            For j:=1 to n do
                If b[i,j]=10000 then Write(f,0:2) Else Write(f,1:2);
            Writeln(f);
        end;
    Close(f);
End;

```

```

BEGIN
    Nhap;

```

XuLy;
InKQ;
END.

Program P21115b; {Tao ma tran lien thong}
Uses Crt;
Const MN=100;
 fn='inp.dat';
 gn='out.dat';
Type mang2=array [1..MN,1..MN] of byte;
 mang1=array [1..MN] of byte;
Var a,b: mang2;
 st,kt: mang1;
 n,s: integer;
 f: text;

Procedure Nhap;
Var i,j: integer;
Begin
 Assign(f,fn); Reset(f);
 Readln(f,n);
 For i:=1 to n do
 begin
 For j:=1 to n do Read(f,a[i,j]);
 Readln(f);
 end;
 Close(f);
End;

Procedure Xuat;
Var i,j: integer;
Begin
 Assign(f,gn); Rewrite(f);
 Writeln(f,n);

```

For i:=1 to n do
  begin
    For j:=1 to n do Write(f,b[i,j], ' ');
    Writeln(f);
  end;
Close(f);
End;
Procedure Xuli;
Var d,i,sst: integer;
Begin
  For s:=1 to n do
    begin
      FillChar(kt,Sizeof(kt),0);
      sst:=1; st[sst]:=s; d:=0;
      While (d<sst) do
        begin
          Inc(d);
          For i:=1 to n do
            If (a[st[d],i]=1)and(kt[i]=0) then
              begin
                Inc(sst);
                st[sst]:=i;
                kt[i]:=1;
                b[s,i]:=1;
              end;
        end;
    end;
End;

BEGIN
  Nhap;
  Xuli;
  Xuat;
END.

```

Bài 12. Bài toán duyệt. Thuật toán quay lui

3. Program P21203; {Liet ke day so}

Uses Crt;

Const out = 'P21203.out';

Var n: integer;

 b: array [1..10] of integer;

 f: text;

Procedure Viet;

Var i: integer;

Begin

 For i:=1 to n do Write(f,b[i]);

 Writeln(f);

End;

Procedure Sinh(i: integer);

Var j: integer;

Begin

 For j:=0 to 4 do

 begin

 b[i]:=j;

 If i=n then Viet

 Else If (i>1) or (j<>0) then Sinh(i+1);

 end;

End;

BEGIN

 Clrscr;

 Write('n = '); Readln(n);

 Assign(f,out); Rewrite(f);

 Sinh(1);

 Close(f);

END.

```

5. Program P21205;
{Liet ke va dem cac tu khac nhau lap tu bang chu cai da cho}
Uses Crt ;
Const  am=['a','e','i'];
        out='P21205.OUT';
Var   c: array[1..5] of char;
        co: boolean;
        count,n: longint;
        f: text;

```

```

Procedure Viet;
Var i: integer;
Begin
  Inc(count);
  For i:=1 to n do Write(f,c[i],');
  Writeln(f);
End;

```

```

Procedure Sinh(i: integer);
Var ch: char; luu: boolean;
Begin
  luu:=co;
  For ch:='a' to 'i' do
    begin
      c[i]:=ch;
      If ch in am then co:=true;
      If (i=n) then begin If co then viet end
      Else sinh(i+1);
      co:=luu;
    end;
  End;

```

```

BEGIN
  Count:=0;

```


1.868

```

Assign(f,out); Rewrite(f);
For n:=2 to 4 do
  begin
    co:=false;
    sinh(1);
  end;
Writeln(f,'So tu = ',count);
Close(f);
END.

```

7. Program P21207; {Liet ke cac hoan vi cua cac to hop m so}

Uses Crt;

Const out = 'P21207.OUT';

Var n,m: Byte;

A: Array [0..100] of Byte;

Cs: Array [1..100] of Byte;

Gn: Array [1..100] of Boolean;

f: Text;

Procedure Viet;

Var i: Byte;

Begin

For i:=1 to m do Write(f,A[Cs[i]],' ');

Writeln(f);

End;

Procedure Chuanbi;

Begin

Clrscr;

Write('Cho so m: '); Readln(m);

Write('Cho so n: '); Readln(n);

Assign(f,out); Rewrite(f);

End;

Procedure SinhHV(i: Byte);

Var j: Byte;

Begin

For j:=1 to m do

 If not Gn[j] then

 begin

 Cs[i]:=j;

 Gn[j]:=True;

 If i=m then Viet

 Else SinhHV(i+1);

 Gn[j]:=False;

 end;

End;

Procedure SinhTH(i: Byte);

Var j: Byte;

Begin

For j:=A[i-1]+1 to n do

 begin

 A[i]:=j;

 If i=m Then

 begin

 FillChar(Gn,SizeOf(Gn),False);

 SinhHV(1)

 end

 Else SinhTH(i+1);

 end;

End;

BEGIN

 Chuanbi;

 A[0]:=0;

 SinhTH(1);

 Close(f);

END.

9. {\$R+,B-}

Program P21209; {Liet ke to hop tap con cua tap da cho}

Uses Crt;

Const out = 'P21209.out';

Var m,n: integer;

a,b: array [0..20] of integer;

cx: array [1..20] of boolean;

f: text;

count: longint;

Procedure Nhap;

Var i,c: integer; S: set of byte;

Begin

Clrscr;

Write('n = '); Readln(n);

Writeln('Nhập một tập n số:');

i:=0; S:=[];

Repeat

Readln(c);

If not(c in S) then

begin

S:=S+[c];

Inc(i);

a[i]:=c;

end;

Until i=n;

End;

Procedure Viet;

Var i: integer;

Begin

Inc(count);

For i:=1 to m do Write(f,a[b[i]],' ');

Writeln(f);

End;

```

Procedure Sinh(i: integer);
Var j: integer;
Begin
  For j:=b[i-1]+1 to n do
    If cx[j] then
      begin
        b[i]:=j;
        cx[j]:=false;
        If i=m then viet
          else sinh(i+1);
        cx[j]:=true;
      end;
  End;

```

```

BEGIN
  Nhap;
  Assign(f,OUT); Rewrite(f);
  count:=0;
  b[0]:=0;
  For m:=1 to n do
    begin
      FillChar(cx,SizeOf(cx),1);
      Sinh(1);
    end;
  Writeln(f,'So cac tap hop con la: ',count);
  Close(f);
END.

```

```

11. Program P21211; {Tim day con don dieu tang dai nhat}
Uses Crt;
Const out = 'P21211.out';
Var  max,n: integer;
     a,b,c: array [0..100] of integer;
     f: text;

```

```

Procedure Nhap;
Var i: integer;
Begin
  ClrScr;
  Write('n = '); Readln(n);
  Writeln('Nhap mot day n so:');
  For i:=1 to n do Read(a[i]);
  Readln;
End;

```

```

Procedure Viet;
Var i: integer;
Begin
  Assign(f,OUT); Rewrite(f);
  Writeln(f,'Day ban dau chua ',n,' phan tu la:');
  For i:=1 to n do Write(f,a[i], ' '); Writeln(f);
  Writeln(f,'Day con dai nhat chua ',Max,' phan tu la:');
  For i:=1 to max do Write(f,a[c[i]], ' '); Writeln(f);
  Close(f);
End;

```

```

Procedure Sinh(i: integer);
Var j,k: integer;
Begin
  For j:=b[i-1]+1 to n do If a[b[i-1]]<a[j] then
    begin
      b[i]:=j;
      If i>Max then
        begin
          Max:=i;
          For k:=1 to Max do c[k]:=b[k];
        end;
      If n>b[i] then sinh(i+1);
    end;
End;

```

```

BEGIN
  Nhap;
  b[0]:=0;
  a[0]:=-MaxInt;
  Max:=0;
  Sinh(1);
  Viet;
END.

```

13. Chứng minh:

* Với $n = 1$ thì $H_1 = 1$ (chuyển đĩa 1 sang cọc 3).

* Với $n = 2$ thì $H_2 = 3$ (chuyển đĩa 1 sang cọc 2, chuyển đĩa 2 sang cọc 3 và chuyển đĩa 1 sang cọc 3).

Rõ ràng $H_2 = 2H_1 + 1$.

* Giả sử công thức đúng đến $n = k$, tức $H_k = 2H_{k-1} + 1$.

Ta phải chứng minh công thức đúng đến $k+1$, tức $H_{k+1} = 2H_k + 1$.

Thật vậy, để chuyển $k+1$ đĩa từ cọc 1 sang cọc 3, ta phải làm lần lượt như sau:

- Chuyển k đĩa phía trên sang cọc 2, hết H_k lần chuyển.
- Chuyển 1 đĩa dưới cùng sang cọc 3, hết 1 lần chuyển.
- Chuyển k đĩa từ cọc 2 sang cọc 3, hết H_k lần chuyển.

Như vậy, tổng cộng ta cần tất cả $2H_k + 1$ lần chuyển - đpcm.

Vậy: $H_n = 2H_{n-1} + 1$ với mọi $n > 1$ (*).

Đặt $H_n = K_n - 1$ thì (*) viết thành $K_n - 1 = 2(K_{n-1} - 1) + 1 \Leftrightarrow K_n = 2K_{n-1}$ với $K_1 = H_1 + 1 = 2$.

Từ đó suy ra $K_n = 2^n$, hay $H_n = 2^n - 1$.

15. {SR+,B-}

Program P21215; {Bài toán xếp ba lô (cai tui)}

Uses Crt;

Const inp = 'P21215.inp';

Var n,m: integer;

```

a,b: array [1..40] of integer;
c: array [0..40,0..500] of word;
mang: array [1..40] of boolean;
f: text;
Procedure Nhap;
Var i: integer;
Begin
  Assign(f,inp); Reset(f);
  Readln(f,n,m);
  For i:=1 to n do Readln(f,a[i],b[i]);
  Close(f);
End;
Procedure Khoitao ;
Begin
  FillChar(c,SizeOf(c),0);
End;
Function max(a,b,c: word): word;
Var m: word;
Begin
  m:=a;
  If b>m then m:=b;
  If c>m then m:=c;
  max:=m;
End;
Procedure qhd;
Var i,j: integer;
Begin
  For i:=1 to n do
    For j:=1 to m do
      If j<a[i] then c[i,j]:=max(c[i,j-1],c[i-1,j],0)
      Else c[i,j]:=max(c[i,j-1],c[i-1,j],c[i-1,j-a[i]]+b[i]);
    End;
  End;

```

```

Procedure Inkq;
Var i,j: integer;
Begin
  Clrscr;
  FillChar(mang,SizeOf(mang),0);
  Writeln('Gia tri lan nhat mang duoc ',c[n,m]);
  i:=n; j:=m;
  While i*j>0 do
    begin
      If c[i,j]=c[i,j-1] then Dec(j)
      Else If c[i,j]=c[i-1,j] then Dec(i)
      Else
        begin
          mang[i]:=true;
          Dec(j,a[i]);
          Dec(i);
        end;
    end;
  Writeln('Danh sach cac vat mang di');
  For i:=1 to n do
    If mang[i] then Write(i:4);
  Readln;
End;

BEGIN
  Nhap;
  Khoitao;
  qhd;
  Inkq;
END.

```

```

17. Program P21217; {Ma tran Latinh}
Uses Crt;
Const out = 'P21217.OUT';

```


Var D,C: Array [1..5] of Byte;
 A: Array [1..5,1..5] of Byte;
 Gn,GnD: Array [1..5] of Boolean;
 f: Text;

Procedure Viet;
 Var i,j,k: Byte;
 Begin
 For i:=1 to 5 do
 begin
 j:=1;
 While C[i]<>D[j] do j:=j+1;
 For k:=0 to 4 do Write(f,D[(i+j+k-2) mod 5+1],' ');
 Writeln(f);
 end;
 Writeln(f);
 End;

Procedure SinhHVD(i: Byte);
 Var j: Byte;
 Begin
 For j:=1 to 5 do
 If Not GnD[j] Then
 begin
 C[i]:=j;
 GnD[j]:=True;
 If i=5 Then Viet Else SinhHVD(i+1);
 GnD[j]:=false;
 end;
 End;

Procedure SinhHV(i: Byte);
 Var j: Byte;
 Begin

```

For j:=1 to 5 do
  If not Gn[j] Then
    begin
      D[i]:=j;
      Gn[j]:=True;
      If i=5 then
        begin
          FillChar(GnD,SizeOf(GnD),false);
          GnD[D[1]]:=True;
          C[1]:=D[1];
          SinhHVD(2);
        end
      Else SinhHV(i + 1);
      GN[j]:=false;
    end;
  End;

```

```

BEGIN
  Assign(f,OUT); Rewrite(f);
  Fillchar(Gn,SizeOf(Gn),false);
  SinhHV(1);
  Close(f);
END.

```

19. {\$B-}

```

Program P21219; {Me cung}
Uses Crt;
Const inp = 'Input.txt';
      out = 'Output.out';
Var  a,pred: array [1..100,1..100] of byte;
      first,last,so: integer;
      m,n,k,l,i,j,dr,cr: byte;
      q: array [1..2,1..10000] of byte;
      f: text;

```

```

Procedure Nhap;
Begin
  Assign(f,inp); Reset(f);
  Readln(f,m,n,k,l);
  For i:=1 to m do
    For j:=1 to n do Read(f,a[i,j]);
  Close(f);
End;

```

```

Procedure Tim;
Begin
  FillChar(pred,SizeOf(pred),0);
  First:=1; Last:=1;
  q[1,first]:=k; q[2,first]:=l;
  pred[k,l]:=1;
  dr:=0; cr:=0;
  While first<=last do
    begin
      i:=q[1,first]; j:=q[2,first];
      Inc(first);
      If (i=1)or(i=m)or(j=1)or(j=n) then
        begin
          dr:=i; cr:=j;
          exit;
        end;
      If (i>1)and(pred[i-1,j]=0)and(a[i-1,j]=0) then
        begin
          pred[i-1,j]:=4;
          If (i-1=1) then begin dr:=i-1; cr:=j; exit; end;
          Inc(last);
          q[1,last]:=i-1; q[2,last]:=j;
        end;
      If (i<m)and(pred[i+1,j]=0)and(a[i+1,j]=0) then

```

```

begin
    pred[i+1,j]:=2;
    If (i+1=m) then begin dr:=i+1; cr:=j; exit; end;
    Inc(last);
    q[1,last]:=i+1; q[2,last]:=j;
end;
If (j>1)and(pred[i,j-1]=0)and(a[i,j-1]=0) then
begin
    pred[i,j-1]:=3;
    If (j-1=1) then begin dr:=i; cr:=j-1; exit; end;
    Inc(last);
    q[1,last]:=i; q[2,last]:=j-1;
end;
If (j<n)and(pred[i,j+1]=0)and(a[i,j+1]=0) then
begin
    pred[i,j+1]:=1;
    If (j+1=n) then begin dr:=i; cr:=j+1; exit; end;
    Inc(last);
    q[1,last]:=i; q[2,last]:=j+1;
end;
end;
End;
Procedure Viet;
Begin
    Assign(f,out); Rewrite(f);
    If dr+cr=0 then Writeln(f,'KHONG')
    Else
        begin
            so:=1;
            Writeln(f,'CO');
            i:=dr; j:=cr; q[1,so]:=i; q[2,so]:=j;
            While (i<>k)or(j<>l) do
                begin
                    If pred[i,j]=1 then

```

```

begin
    Dec(j); Inc(so);
    q[1,so]:=i; q[2,so]:=j;
end;
If pred[i,j]=2 then
begin
    Dec(i); inc(so);
    q[1,so]:=i; q[2,so]:=j;
end;
If pred[i,j]=3 then
begin
    Inc(j); Inc(so);
    q[1,so]:=i; q[2,so]:=j;
end;
If pred[i,j]=4 then
begin
    Inc(i); Inc(so);
    q[1,so]:=i; q[2,so]:=j;
end;
end;
For i:=so downto 1 do Writeln(f,q[1,i]:4,q[2,i]:4);
end;
Close(f);
End;

BEGIN
Nhap;
Tim;
Viet;
END.

```

Bài 13. Đường đi Euler

1. Trong đồ thị 1 đường đi a, b, c, d, a là đường đi Euler vì nó đi qua mỗi cạnh của đồ thị đúng 1 lần, còn đường đi a, b, c, d không phải là đường đi Euler vì nó không đi qua cạnh (d, a).

Tương tự như vậy ta có thể lí giải cho các đường đi còn lại.

3. Một đường đi tiết kiệm nhất hiển nhiên phải là một chu trình Euler vì nó đi qua mỗi cạnh chỉ đúng 1 lần. Có thể đi theo đường sau: P, F, C, B, A, D, G, H, P, B, D, C, E, G, B, E, H, F, E, D, P.

5. Một đồ thị vô hướng liên thông có đường đi Euler khi và chỉ khi số các đỉnh có bậc lẻ không vượt quá 2 (xem thêm câu 9).

7. Program P21307; {Tìm chu trình Euler}

Uses Crt;

Const inp='P21308.inp';

u=1;

Var a: array [1..100,1..100] of byte;

n: byte;

f: text;

Procedure Input;

Var i,j: byte;

Begin

Assign(f,inp); Reset(f);

Readln(f,n);

For i:=1 to n do

begin

For j:=1 to n do Read(f,a[i,j]);

Readln(f);

end;

Close(f);

End;

```

Function Kiemtra: boolean;
Var i,j,s,d: byte;
Begin
  d:=0;
  For i:=1 to n do
    begin
      s:=0;
      For j:=1 to n do s:=s+a[i,j];
      If Odd(s) then Inc(d);
    end;
  If d>0 then Kiemtra:=false Else Kiemtra:=true;
End;

```

```

Procedure Tim;
Var v,x,top,dCE: byte;
    Stack,CE: array [1..100] of byte;
Begin
  top:=1; Stack[top]:=u; dCE:=0;
  Repeat
    v:=Stack[top];
    x:=1;
    While (x<=n)and(a[v,x]=0) do Inc(x);
    If x>n then begin Inc(dCE); CE[dCE]:=v; Dec(top); end
    Else
      begin
        Inc(top); Stack[top]:=x;
        a[v,x]:=0; a[x,v]:=0;
      end;
  Until top=0;
  Writeln('Co chu trinh Euler:');
  For x:=dCE downto 1 do write(CE[x], ' ');
End;

```

BEGIN
 Clrscr;
 Input;
 If Kiemtra then Tim
 Else Write('Khong co chu trinh Euler.');

Readln;

END.

9. Program P21309; {Tim duong di Euler}

Uses Crt;

Const inp='P21309.inp';

Var a: array [1..100,1..100] of byte;

 n,u: byte;

 f: text;

Procedure Input;

Var i,j: byte;

Begin

Assign(f,inp); Reset(f);

Readln(f,n);

For i:=1 to n do

 begin

 For j:=1 to n do Read(f,a[i,j]);

 Readln(f);

 end;

Close(f);

End;

Function Kiemtra: boolean;

Var i,j,s,d: byte;

Begin

 d:=0; u:=1;

 For i:=1 to n do

 begin

 s:=0;


```

    For j:=1 to n do s:=s+a[i,j];
    If Odd(s) then begin Inc(d); u:=i; end;
end;
If d>2 then Kiemtra:=false Else Kiemtra:=true;
End;

```

```

Procedure Tim;
Var v,x,top,dCE: byte;
    Stack,CE: array [1..100] of byte;
Begin
    top:=1; Stack[top]:=u; dCE:=0;
    Repeat
        v:=Stack[top];
        x:=1; While (x<=n)and(a[v,x]=0) do Inc(x);
        If x>n then begin Inc(dCE); CE[dCE]:=v; Dec(top); end
        Else
            begin
                Inc(top); Stack[top]:=x;
                a[v,x]:=0; a[x,v]:=0;
            end;
    Until top=0;
    Writeln('Co duong di Euler:');
    For x:=dCE downto 1 do write(CE[x],' ');
End;

```

```

BEGIN
    Clrscr;
    Input;
    If Kiemtra then Tim
    Else Write('Khong co duong di Euler. ');
    Readln;
END.

```

11. Program P21311; {Liet ke cac duong di Euler}

Uses Crt;

Const, inp='P21308.inp';

out='P21311.out';

Type M1=array [0..10] of byte;

M2=array [1..10,1..10] of byte;

Var a: M2;

b: M1;

n,m,i,u: byte;

f: text;

Ok: boolean;

Procedure Input;

Var i,j,s,d: byte;

Begin

Assign(f,inp); Reset(f);

Readln(f,n);

u:=1; m:=0; d:=0;

For i:=1 to n do

begin

s:=0;

For j:=1 to n do

begin

Read(f,a[i,j]);

s:=s+a[i,j];

end;

If Odd(s) then begin Inc(d); u:=i; end;

m:=m+s;

Readln(f);

end;

m:=m div 2;

Close(f);

If d>2 then Ok:=false Else Ok:=true;

End;

Procedure InB;

Var i: integer;

```

Begin
  For i:=0 to m do Write(f,b[i],' ');
  Writeln(f);
end;
Procedure Tim(CE: M1; a:M2; i:byte);
Var j,k: byte;
Begin
  For j:=1 to n do
    If (a[b[i-1],j]=1) then
      begin
        a[b[i-1],j]:=0; a[j,b[i-1]]:=0;
        b[i]:=j;
        If i=m then InB
        Else Tim(b,a,i+1);
        a[b[i-1],j]:=1; a[j,b[i-1]]:=1;
      end;
  End;
BEGIN
  Clrscr;
  Input;
  b[0]:=u; i:=1;
  Assign(f,out); Rewrite(f);
  If Ok then Tim(b,a,i)
  Else Write('Khong co duong di Euler. ');
  Close(f);
  Readln;
END.

```

13. a. Với mọi $n \geq 3$.

b. Giống câu a.

c. $v_1, v_2, \dots, v_n$.

15. a. Không có vì tất cả các đỉnh (trừ tâm) đều là bậc 3 - lẻ.

b. Giống câu a.

Bài 14. Chu trình Hamilton

1. Trong đồ thị 1, đường đi a, b, c, d là đường đi Hamilton vì nó đi qua mỗi đỉnh đúng 1 lần, còn đường đi a, b, c không phải là đường đi Hamilton vì nó không đi qua đỉnh d.

Trong đồ thị 3, đường đi a, b, c, d, a, c không phải là đường đi Hamilton vì nó đi qua đỉnh a (và đỉnh c) 2 lần ...

3. Program P21403; {Liet ke cac chu trinh Hamilton}

Uses Crt;

Const inp='P21403.inp';

out='P21403.out';

Type M1=array [0..10] of byte;

M2=array [1..10,1..10] of byte;

Var a: M2;

b,c: M1;

n,i,d: byte;

f: text;

Procedure Input;

Var i,j: byte;

Begin

Assign(f,inp); Reset(f);

Readln(f,n);

For i:=1 to n do

begin

For j:=1 to n do Read(f,a[i,j]);

Readln(f);

end;

Close(f);

For i:=1 to n do c[i]:=0;

End;

```

Procedure InB(b: M1);
Var i: integer;
Begin
  For i:=0 to n do Write(f,b[i], ' '); Writeln(f);
  Inc(d);
end;

```

```

Procedure Tim(b,c: M1; i:byte);
Var j,k: byte;
Begin
  For j:=1 to n do
    If (a[b[i-1],j]=1)and(c[j]=0) then
      begin
        b[i]:=j; c[j]:=1;
        If i<n then Tim(b,c,i+1)
        Else If b[i]=b[0] then InB(b);
        c[j]:=0;
      end;
  end;
End;

```

```

BEGIN
  Input;
  b[0]:=1; i:=1; d:=0;
  Assign(f,out); Rewrite(f);
  Tim(b,c,i);
  If d=0 then Write(f,'Khong co chu trinh Hamilton. ');
  Close(f);
END.

```

5. Program P21405; {Liet ke cac duong di Hamilton}

Uses Crt;

Const inp='P21404.inp';

out='P21404.out';

Type M1=array [1..50] of byte;

```

    M2=array [1..50,1..50] of byte;
Var  a: M2;
     b,c: M1;
     n,i,d: byte;
     f: text;

```

```

Procedure Input;
Var i,j: byte;
Begin
  Assign(f,inp); Reset(f);
  Readln(f,n);
  For i:=1 to n do
    begin
      For j:=1 to n do Read(f,a[i,j]);
      Readln(f);
    end;
  Close(f);
  For i:=1 to n do c[i]:=0;
End;

```

```

Procedure InB(b: M1);
Var i: integer;
Begin
  For i:=1 to n do Write(f,b[i], ' '); Writeln(f);
  Inc(d);
End;

```

```

Procedure Tim(b,c: M1; i:byte);
Var j,k: byte;
Begin
  For j:=1 to n do
    If (a[b[i-1],j]=1)and(c[j]=0) then
      begin
        b[i]:=j; c[j]:=1;

```

```

        If i<n then Tim(b,c,i+1)
        Else InB(b);
        c[j]:=0;
    end;
End;

BEGIN
    Input;
    b[1]:=1; i:=2; d:=0;
    Assign(f,out); Rewrite(f);
    Tim(b,c,i);
    If d=0 then Write(f,'Khong co duong di Hamilton.');
```

Close(f);

END.

7. Hiển nhiên vì với $n > 1$ thì đồ thị có nhiều hơn $2^1 = 2$ đỉnh và mọi đỉnh của đồ thị đều có bậc $n > \frac{n}{2}$ nên theo định lí ở câu 5 suy ra đồ thị có chu trình Hamilton.

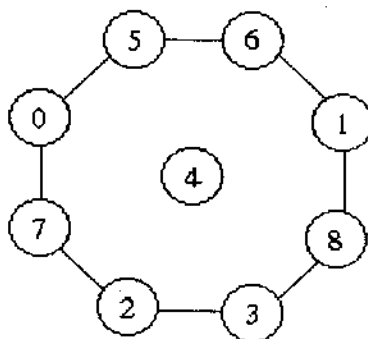
9. a, b. Luôn có với mọi n .

c. Tự tìm chu trình (đường) Hamilton của đồ thị.

11.

6	7	8
3	4	5
0	1	2

Bàn cờ



Đồ thị

13. Trên bàn cờ ta chia các ô thành 2 nhóm: V_1 gồm các ô (x,y) với $x+y$ chẵn và V_2 gồm các ô (x,y) với $x+y$ lẻ. Rõ ràng trong đồ thị mô phỏng đường đi của con mã trên bàn cờ không thể có 1 cạnh nào nối 2 đỉnh nằm trong 1 tập V_1 hoặc V_2 . Do đó, đồ thị mô phỏng chuyển động của con mã trên bàn cờ là đồ thị kép.

15. Program P21415; {Tìm 1 đường đi khép kín của con mã}

Uses Crt;

Const out = 'P21415.out';

n=8;

Type M1=array [0..n*n-1] of byte;

M2=array [0..n*n-1,0..n*n-1] of byte;

Var a: M2;

b,c: M1;

i: byte;

f: text;

Procedure Input;

Var i,j,xi,yi,xj,yj: byte;

Begin

For i:=0 to n*n-1 do

For j:=0 to n*n-1 do

begin

xi:=i div n; yi:=i mod n;

xj:=j div n; yj:=j mod n;

If ((abs(xi-xj)=1)and(abs(yi-yj)=2))or ((abs(xi-xj)=2)and(abs(yi-yj)=1)) then a[i,j]:=1

Else a[i,j]:=0;

end;

For i:=0 to n*n-1 do c[i]:=0;

End;

Procedure InB(b: M1);

Var i: integer;

Begin


```

For i:=0 to n*n-1 do Write(f,b[i],' '); Writeln(f);
Close(f);
Halt;
End;

```

```

Procedure Tim(b,c: M1; i:byte);
Var j,k: byte;
Begin
For j:=0 to n*n-1 do
If (a[b[i-1],j]=1)and(c[j]=0) then
begin
b[i]:=j; c[j]:=1;
If i<n*n-1 then Tim(b,c,i+1)
Else If (a[b[i],b[0]]=1) then InB
c[j]:=0;
end;
End;

```

```

BEGIN
Input;
b[0]:=0; c[0]:=1; i:=1;
Assign(f,out); Rewrite(f);
Write('...');
Tim(b,c,i);
Close(f);
END.

```

17. Giả sử Kanguru đứng ở ô (0,0).

Ta sẽ chứng minh Kanguru không thể đi tới ô (1,0) bằng phản chứng.

Thực vậy, giả sử sau m bước (+1,-1) và n bước (-5,+7), Kanguru đi tới ô (0,1). Khi đó, ta có $0 = 1m - 5n$ và $1 = -1m + 7n$, suy ra

$1 = 2n$ hay $n = \frac{1}{2}$ là điều vô lý vì n là số nguyên.

Vậy từ một điểm ban đầu bất kì, Kanguru không bao bao giờ có thể đi khắp các ô của lưới, ngay cả khi cho phép đi qua 1 ô nhiều lần (tất nhiên phải loại trừ trường hợp $N = 1$).

19. Program P21419; {Xep domino}

Uses Crt;

Var a: array [1..28,1..2] of byte;

dx: array [1..28] of byte;

b: array [1..100] of byte;

i,j,dem: byte;

Procedure Tim(k: byte);

Var i: byte;

Begin

If k>28 then

begin

For i:=1 to 28 do Write(b[(i-1)*2+1], ' ', b[(i-1)*2+2], ' ');

Readln;

Halt;

end;

For i:=1 to 28 do

If (dx[i]=0) then

begin

dx[i]:=1;

If a[i,1]=b[2*(k-1)] then

begin

b[2*k-1]:=a[i,1];

b[2*k]:=a[i,2];

Tim(k+1);

end;

If (a[i,1]<>a[i,2])and(a[i,2]=b[2*(k-1)]) then

begin

b[2*k-1]:=a[i,2];

b[2*k]:=a[i,1];

```

        Tim(k+1);
    end;
    dx[i]:=0;
end;
End;

BEGIN
    ClrScr;
    dem:=0;
    For i:=0 to 6 do
        For j:=i to 6 do
            begin
                Inc(dem);
                a[dem,1]:=i;
                a[dem,2]:=j;
            end;
        FillChar(dx,sizeof(dx),0);
        For i:=1 to 28 do
            begin
                FillChar(b,Sizeof(b),0);
                dx[i]:=1;
                b[1]:=a[i,1];
                b[2]:=a[i,2];
                Tim(2);
                dx[i]:=0;
            end;
        END.

```

Bài 15. Các bài toán trên đồ thị

1. Program P21501; {Ve Do thi bang 1 net}

Uses Crt;

Const inp='P21501.inp';

Var a: array [1..100,1..100] of byte;

n,u: byte;

f: text;

Procedure Input;

Var i,j: byte;

Begin

Assign(f,inp); Reset(f);

Readln(f,n);

For i:=1 to n do

begin

For j:=1 to n do Read(f,a[i,j]);

Readln(f);

end;

Close(f);

End;

Function Kiemtra: boolean;

Var i,j,s,d: byte;

Begin

d:=0; u:=1;

For i:=1 to n do

begin

s:=0; For j:=1 to n do s:=s+a[i,j];

If Odd(s) then begin Inc(d); u:=i; end;

end;

If d>2 then Kiemtra:=false Else Kiemtra:=true;

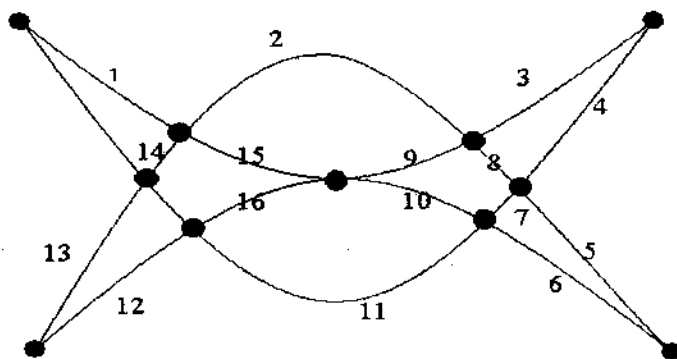
End;

```

Procedure Tim;
Var v,x,top,dCE: byte;
    Stack,CE: array [1..100] of byte;
Begin
top:=1; Stack[top]:=u; dCE:=0;
Repeat
    v:=Stack[top];
    x:=1; While (x<=n)and(a[v,x]=0) do Inc(x);
    If x>n then begin Inc(dCE); CE[dCE]:=v; Dec(top); end
    Else
        begin
            Inc(top); Stack[top]:=x;
            a[v,x]:=0; a[x,v]:=0;
        end;
Until top=0;
Writeln('Do thi la "don lien".');
Writeln('Co the ve 1 net theo cac dinh sau:');
For x:=dCE downto 1 do write(CE[x],' ');
End;

BEGIN
Clrscr;
Input;
If Kiemtra then Tim
    Else Write('Do thi khong phai la "don lien".');
Readln;
END.

```



```

Program P21503;
Uses Crt;
Const  inp='P21503.inp';
       d:array[1..4]of shortint=(-1,0,1,0);
       c:array[1..4]of shortint=(0,-1,0,1);
Var  f:text;
     m,n,dxp,cxp,dt,ct:byte;
     a,b,pre :array[0..51,0..51]of byte;
     qd,qc :array[1..2500]of byte;

```

```

Procedure Nhap;
Var i,j: byte;
Begin
Assign(f,inp);Reset(f);
Readln(f,m,n);
For i:=1 to m do
  begin
    For j:=1 to n do Read(f,a[i,j]);
    Readln(f);
  end;
Close(f);

```

```

Repeat
  Writeln('Nhap toa do diem xuat phat tu do: ');
  Write(' - Dong: '); Readln(dxp);
  Write(' - Cot : '); Readln(cxp);
Until a[dxp,cxp]=0;
End;

```

```

Procedure Chuan_bi;
Var i,j: byte;
Begin
  For i:=0 to m+1 do
    begin
      a[i,0]:=2;
      a[i,n+1]:=2;
    end;
  For i:=0 to n+1 do
    begin
      a[0,i]:=2;
      a[m+1,i]:=2;
    end;
  FillChar(b,Sizeof(b),0);
  pre[dxp,cxp]:=0;
  b[dxp,cxp]:=1;
  dt:=0;
  ct:=0;
End;

```

```

Procedure BFS;
Var i,j,dau,cuoi,dong,cot: byte;
Begin
  dau:=1; cuoi:=1;
  qd[1]:=dxp; qc[1]:=cxp;
  While dau<=cuoi do
    begin

```

```

    dong:=qd[dau];
    cot:=qc[dau];
    Inc(dau);
    For i:=1 to 4 do
        begin
            If (a[dong+d[i],cot+c[i]]=2) then
                begin
                    dt:=dong;
                    ct:=cot;
                    exit;
                end;
            If (a[dong+d[i],cot+c[i]]=0)and(b[dong+d[i],cot+c[i]]=0)
then
                begin
                    b[dong+d[i],cot+c[i]]:=1;
                    pre[dong+d[i],cot+c[i]]:=i;
                    Inc(cuoi);
                    qd[cuoi]:=dong+d[i];
                    qc[cuoi]:=cot+c[i];
                end;
            end;
        end;
    End;

```

```

Procedure Tim_duong(i,j:byte);
Begin
    If pre[i,j]<>0 then
        Case pre[i,j] of
            1:Tim_duong(i+1,j);
            2:Tim_duong(i,j+1);
            3:Tim_duong(i-1,j);
            4:Tim_duong(i,j-1);
        end;
    Write('[' ,i ,',' ,j ,'] ');Readln;

```



```

End;

Procedure Xu_ly;
Begin
Chuan_bi;
BFS;
If dt<>0 then
    begin
        Writeln('Trinh tu duong di:');
        Tim_duong(dt,ct);
        exit;
    end
Else Writeln('Khong the thuc hien di chuyen ra ngoai');Readln;
End;

```

```

BEGIN
  Clrscr;
  Nhap;
  Xu_ly;
  Readln;
END.

```

5. Program P21505; {Tim duong di tu s den t}

```

Uses Crt;
Const N=64;
      fn='P21505.INP';
      gn='P21505.OUT';
Type mang2=array [1..N,1..N] of byte;
      mang1=array [0..N] of byte;
Var a,e: mang2;
    Qu,kt,dd,ddd: mang1;
    s,ss,t,sst,xi,xj,yi,yj,k: integer;
    f,g: text;
    Co: boolean;

```

```

Procedure Nhap;
Var i,j,ds,dt,cs,ct: integer;
Begin
Assign(f,fn); Reset(f);
For i:=1 to 8 do
begin
    For j:=1 to 8 do Read(f,e[i,j]);
    Readln(f);
end;
Close(f);
Clrscr;
For i:=1 to 8 do
begin
    For j:=1 to 8 do
begin
        GotoXY(2*i+20,j+4);
        Write(e[i,j]:2);
end;
end;
GotoXY(1,1);
Write('ds,cs = '); Readln(cs,ds);
Write('dt,ct = '); Readln(ct,dt);
s:=(ds-1)*8+cs; t:=(dt-1)*8+ct;
TextColor(4);
GotoXY(2*ds+20,cs+4); Write('*':2);
GotoXY(2*dt+20,ct+4); Write('*':2);
Readln;
ss:=s;
For i:=1 to n do
    For j:=1 to n do
begin
    xi:=(i-1)div 8+1; yi:=(i-1)mod 8+1;
    xj:=(j-1)div 8+1; yj:=(j-1)mod 8+1;

```

63

```

        If ((xi=xj) and (abs(yi-yj)=1) and (e[xi,yi]=0) and
(e[xj,yj]=0)) or ((yi=yj)
        and (abs(xi-xj)=1) and (e[xi,yi]=0) and
(e[xj,yj]=0)) then a[i,j]:=1
        Else a[i,j]:=0;
    end;
End;

```

```

Procedure Init;
Begin
    FillChar(kt,SizeOf(kt),0);
    FillChar(dd,SizeOf(dd),0);
    sst:=1; Qu[sst]:=s; kt[s]:=1;
End;

```

```

Procedure Tim;
Var d,i,j: integer;
Begin
    d:=0; Co:=false;
    While (d<sst)and(not Co) do
        begin
            Inc(d); s:=Qu[d];
            For i:=1 to n do
                If (a[s,i]=1)and(kt[i]=0) then
                    begin
                        Inc(sst); Qu[sst]:=i;
                        kt[i]:=1;
                        dd[sst]:=s;
                        If (i=t) then Co:=true;
                    end;
            end;
            k:=0;
            If Co then
                begin

```

```

    i:=t;
  Repeat
    j:=1; While qu[j]<>i do Inc(j);
    Inc(k); ddd[k]:=dd[j];
    i:=dd[j];
  Until i=ss;
  ddd[0]:=t;
end
Else Writeln('Khong co duong di.');
```

End;

```

Procedure Viet;
Var i: integer;
Begin
  TextColor(4);
  For i:=1 to k-1 do
    begin
      xi:=(ddd[i]-1)div 8+1;
      yi:=(ddd[i]-1)mod 8+1;
      GotoXY(2*xi+20,yi+4); Write('*':2);
    end;
  Readln;
End;
```

```

BEGIN
  Clrscr;
  Nhap;
  Init;
  Tim;
  Viet;
  Readln;
END.
```

7. Program P21507;

{Duong di ngan nhat tu A den B chi di tren cac canh cua luoi}

Uses Crt;

Var m,n: byte;

 a: array [1..26,1..26] of longint;

Procedure Chuan_bi;

Var i: byte;

Begin

 Write('Do rong cua luoi (khong qua 25) M = ');Readln(m);

 Write('Do dai cua luoi (khong qua 25) N = ');Readln(n);

 For i:=1 to m+1 do a[i,1]:=1;

 For i:=1 to n+1 do a[1,i]:=1;

End;

Procedure Xu_ly;

Var i,j: byte;

Begin

 For i:=2 to m+1 do

 For j:=2 to n+1 do a[i,j]:=a[i,j-1]+a[i-1,j];

 Writeln('So cac duong di ngan nhat tu A den B: ',a[m+1,n+1]);

 Readln;

End;

BEGIN

 ClrScr;

 Chuan_bi;

 Xu_ly;

END.

9. Program P21509; {Tim do dai duong di ngan nhat}

Uses Crt;

Const nmax=100;

 inp='INPUT.DAT';

Var f: text;

```

n,imin: byte;
d,dmin: longint;
a: array [1..nmax,1..nmax] of integer;
b,bmin,c: array [1..nmax] of byte;

```

```

Procedure Nhap;
Var i,j: byte;
Begin
Assign(f,inp); Reset(f);
Readln(f,n);
For i:=1 to n do
begin
For j:=1 to n do Read(f,a[i,j]);
Readln(f);
end;
Close(f);
End;

```

```

Procedure Viet;
Var i: integer;
Begin
Writeln('Duong di ngan nhat la:');
For i:=1 to imin do Write(bmin[i],'); Writeln;
Write('Co do dai ',dmin);
Readln;
End;

```

```

Procedure Sinh(i: integer);
Var j,k: integer;
Begin
For j:=1 to n do
If (a[b[i-1],j]<>0)and(c[j]=0)and(d<dmin) then
begin
b[i]:=j; c[j]:=1; d:=d+a[b[i-1],j];

```

059.

```

    If (j=n) then
      begin
        If dmin>d then
          begin
            dmin:=d;
            imin:=i;
            For k:=1 to i do bmin[k]:=b[k];
          end;
        end
      Else If (i<n) then sinh(i+1);
      c[j]:=0; d:=d-a[b[i-1]j];
    end;
  End;

```

```

BEGIN
  Clrscr;
  Nhap;
  b[1]:=1; dmin:=MaxInt;
  Sinh(2);
  Viet;
  Readln;
END.

```

11. Program P21511; {Chia luoi}

```

Uses crt;
Const fi = 'luoi.dat';
      fo = 'luoi.out';
      px:array[1..4] of shortint = (0,1,0,-1);
      py:array[1..4] of shortint = (-1,0,1,0);
Var f:text;
    m,n:byte;
    a,c,b:array[1..10,1..10] of byte;
    min,tong,tong2:integer;
Procedure input;

```

```

Var i,j:byte;
Begin
  Assign(f,fi);
  Reset(f);
  Readln(f,m,n);
  tong:=0;
  For i:=1 to m do
    Begin
      For j:=1 to n do
        Begin
          Read(f,a[i,j]);
          tong:=tong+a[i,j];
        End;
      Readln(f);
    End;
  close(f);
End;

Function test(x,y:byte):boolean;
Var x1,y1,i:byte;
Begin
  test:=true;
  For i:=1 to 4 do
    Begin
      x1:=x+px[i];
      y1:=y+py[i];
      If (x1 in [1..n]) and (y1 in [1..m]) and (b[y1,x1]=1) then
        exit;
    End;
  test:=false;
End;

Procedure tim(sum:integer);
Var i,j:byte;

```



```

Begin
  If abs(tong2-sum)<min then
    Begin
      c:=b;
      min:=tong2-sum;
    End;
  If sum>tong2 then exit;
  For i:=1 to m do
    For j:=1 to n do
      If (b[i,j]=0) and test(j,i) then
        Begin
          b[i,j]:=1;
          tim(sum+a[i,j]);
          b[i,j]:=0;
        End;
    End;
  End;

Procedure work;
Begin
  Fillchar(b,sizeof(b),0);
  b[1,1]:=1;
  tong2:=tong div 2;
  min:=tong2;
  tim(a[1,1]);
End;

Procedure output;
Var i,j:byte;
Begin
  Assign(f,fo);
  ReWrite(f);
  For i:=1 to m do
    Begin
      For j:=1 to n do Write(f,' ',c[i,j]);
    End;
  End;

```

```

        Writeln(f);
    End;
    Close(f);
End;

```

```

BEGIN
    input;
    work;
    output;
END.

```

13. Program P21513; {Xay dung duong di giua cac thanh pho}

```

Uses Crt;
Const  nmax=20;
Type  M2=array [1..nmax,1..nmax] of 0..1;
Var f: text;
    m,n,dmin: byte;
    a,amin: M2;

```

```

Procedure Nhap;
Begin
    Write(' N = '); Readln(n);
    m:=n*(n-1) div 2;
End;

```

```

Procedure Viet;
Var r,s: integer;
Begin
    Writeln('Ma tran duong di tot nhat la:');
    For r:=1 to n do
        begin For s:=1 to n do Write(amin[r,s]:2); Writeln; end;
    Write('So duong di la ',dmin);
    Readln;
End;

```

Function OK(a: M2): boolean;
 Var i,j,k: integer;
 Kt1,Kt2: boolean;
 Begin
 Kt1:=true;
 For i:=1 to n do For j:=1 to n do
 If $i < j$ then
 begin
 Kt2:=false;
 For k:=1 to n do
 If $(a[i,j]=1) \text{ or } ((a[i,k]=1) \text{ and } (a[k,j]=1))$ then Kt2:=true;
 If Kt2=false then Kt1:=false;
 end;
 OK:=Kt1;
 End;

Function d(a: M2): integer;
 Var i,j,k: integer;
 Begin
 k:=0;
 For i:=2 to n do
 For j:=1 to i-1 do If $a[i,j]=1$ then Inc(k);
 d:=k;
 End;

Procedure Sinh(i: integer);
 Var j,r,s: integer;
 Begin
 For j:=0 to 1 do
 begin
 r:=2; While not($i \leq r * (r-1) \text{ div } 2$) do Inc(r);
 s:= $i - ((r-1) * (r-2) \text{ div } 2)$;
 a[r,s]:=j; a[s,r]:=j;

```

If (i<m)and(d(a)<dmin) then sinh(i+1)
Else If (i=m)and(dmin>d(a))and OK(a) then
    begin
        dmin:=d(a);
        For r:=1 to n do
            For s:=1 to n do amin[r,s]:=a[r,s];
        end;
    end;
end;
End;

```

```

BEGIN
    Clrscr;
    Nhap;
    dmin:=255;
    Sinh(1);
    Viet;
    Readln;
END.

```

15. Program P21515; {Sen bo tren hcn M*N}

```

Uses Crt, Graph;
Const M=2; N=10;
Type mang=array[1..2,0..M+N] of byte;
Var a,b: mang;
    Gd,Gm,k,i,W,x0,y0,d:integer;
    S: string;

```

```

Function Khac(a,b: mang): boolean;
Var kh: boolean; i: byte;
Begin
    kh:=false;
    For i:=1 to k do
        If (a[1,i]<>b[1,i])or(a[2,i]<>b[2,i]) then kh:=true;
    Khac:=kh;

```

End;

Procedure Ve(Var a: mang);

Var i,j,x,y: integer;

Begin

For j:=0 to M do

For i:=0 to N do

begin

x:=i*W+x0; y:=y0-j*W;

Bar(x-1,y-1,x+1,y+1);

end;

OutTextXY(GetMaxX div 2,GetMaxY-10,S);

SetColor(4);

Str(d,S); OutTextXY(GetMaxX div 2,GetMaxY-10,S); Inc(d);

For i:=1 to k do

begin

Line(W*a[2,i-1]+x0,y0-a[1,i-1]*W,a[2,i]*W+x0,y0-a[1,i]*W);

Delay(20);

end;

SetColor(16);

For i:=1 to k do

begin

Line(a[2,i-1]*W+x0,y0-a[1,i-1]*W,a[2,i]*W+x0,y0-a[1,i]*W);

end;

End;

Procedure Tiep(Var a: mang);

Var i,j: byte;

Begin

i:=k;

While a[1,i]=a[1,i-1] do dec(i);

While a[2,i]=a[2,i-1] do dec(i);

```

Inc(a[1,i]); Dec(a[2,i]);
For j:=i+1 to k do
  If a[2,j-1]<N then begin a[2,j]:=a[2,j-1]+1; a[1,j]:=a[1,i]; end
  Else begin a[2,j]:=N; a[1,j]:=a[1,j-1]+1; end;
Ve(a);
End;

```

```

BEGIN
  Gd:=Detect;
  InitGraph(Gd,Gm,' ');
  y0:=GetMaxY-50;
  x0:=100;
  W:=(GetMaxX-2*x0) div N;
  If (GetMaxX-2*x0) div N>(y0-10) div M then W:=(y0-10) div
M;
  k:=M+N;
  For i:=0 to N do begin a[1,i]:=0; a[2,i]:=i; end;
  For i:=N+1 to k do begin a[1,i]:=i-N; a[2,i]:=N; end; {Dau}
  For i:=0 to M do begin b[1,i]:=i; b[2,i]:=0; end;
  For i:=M+1 to k do begin b[1,i]:=M; b[2,i]:=i-M; end; {Cuoi}
  d:=1; Ve(a);
  While Khac(a,b) do Tiep(a);
  Readln;
  CloseGraph;
END.

```

17. Program P21517;

Uses Crt;

Var n:byte;

a,b:array[1..100,1..100] of byte;

d,c:array[1..100] of byte;

count:word;

Procedure input;

Begin
 Write('Nhap n = ');Readln(n);
 n:=2*n;
 End;

Function dc(x:byte):byte;
 Var i,d:byte;
 Begin
 d:=0;
 For i:=1 to n do
 If a[i,x]=1 then inc(d);
 dc:=d;
 End;

Function dd(y:byte):byte;
 Var d,i:byte;
 Begin
 d:=0;
 For i:=1 to n do
 If a[y,i]=1 then inc(d);
 dd:=d;
 End;

Procedure output;
 Var x,y:byte;
 Begin
 For y:=1 to n do If dd(y)=1 then break;
 For x:=1 to n do If dc(x)=1 then break;
 b:=a;
 b[y,x]:=1;
 For y:=1 to n do
 For x:=1 to n do
 Begin
 Gotoxy(x*2,y);

```

        Write(b[y,x]);
    End;
    delay(100);
End;

```

```

Procedure hien;
Var i,j:byte;
Begin
    If count=2*n - 1 then output;
End;

```

```

Procedure timd(y:byte); Forward;

```

```

Procedure timc(x:byte);
Var i:byte;
Begin
    hien;
    For i:=1 to n do
        If d[i]=0 then
            Begin
                d[i]:=1;
                a[i,x]:=1;
                c[x]:=1;
                inc(count);
                timd(i);
                dec(count);
                c[x]:=0;
                a[i,x]:=0;
                d[i]:=0;
            End;
    End;
End;

```

```

Procedure timd(y:byte);
Var i:byte;

```



```

Begin
  hien;
  For i:=1 to n do
    If c[i]=0 then
      Begin
        c[i]:=1;
        d[y]:=1;
        a[y,i]:=1;
        inc(count);
        timc(i);
        dec(count);
        a[y,i]:=0;
        d[y]:=0;
        c[i]:=0;
      End;
    End;
  End;

```

```

Procedure work;
Var i:byte;
Begin
  fillchar(a,sizeof(a),0);
  fillchar(c,sizeof(c),0);
  fillchar(d,sizeof(d),0);
  count:=0;
  timd(1);
End;

```

```

BEGIN
  input;
  clrscr;
  work;
  Readln;
END.

```

M Ụ C L Ụ C

Lời Nhà xuất bản	3
Lời nói đầu	5
PHẦN I. TÓM TẮT LÝ THUYẾT VÀ ĐỀ BÀI TẬP	
Bài 1. Mạng nhiều chiều	7
Bài 2. Các thao tác trên xâu kí tự	11
Bài 3. Kiểu tập hợp	15
Bài 4. Mô phỏng đồ thị	20
Bài 5. Kiểu bản ghi	28
Bài 6. Kiểu file	33
Bài 7. Kiểu con trỏ	39
Bài 8. Vẽ đồ thị hàm số	46
Bài 9. Các thuật toán tạo chuyển động trên đồ họa	48
Bài 10. Các bài toán hình học	52
Bài 11. Ôn tập	56
Bài 12. Bài toán duyệt. Thuật toán quay lui	59
Bài 13. Đường đi Euler	69
Bài 14. Chu trình Hamilton	76
Bài 15. Các bài toán trên đồ thị	80
PHẦN II. CH ƯƠNG TRÌNH	87

Chịu trách nhiệm xuất bản :

Chủ tịch HĐQT kiêm Tổng Giám đốc **NGÔ TRẦN ÁI**
Phó Tổng Giám đốc kiêm Tổng biên tập **NGUYỄN QUÝ THAO**

Biên tập nội dung:

NGÔ ÁNH TUYẾT

Trình bày bìa:

TẠ TRỌNG TRÍ

Chế bản:

TRẦN BÍCH VÂN

LẬP TRÌNH PASCAL - TẬP II

Mã số : 8II785T6 - DAI

In 1.000 cuốn khổ 14,5cm x 20,5cm. Tại **CÔNG TY CỔ PHẦN IN ANH VIỆT**
Số 74 ngõ 310 đường Nghi Tàm - Tây Hồ - Hà Nội.
Giấy phép xuất bản : 04 - 2006/CXB/193 - 1860/GD.
In xong và nộp lưu chiểu Quý I năm 2006.



CÔNG TY CỔ PHẦN SÁCH ĐẠI HỌC – DẠY NGHỀ

HEVOBCO

Địa chỉ : 25 Hàn Thuyên, Hà Nội

TÌM ĐỌC

TỦ SÁCH "TIN HỌC TRONG NHÀ TRƯỜNG "

CỦA NHÀ XUẤT BẢN GIÁO DỤC

- 1. Một số vấn đề chọn lọc trong môn tin học** (tập một, tập hai)
Nguyễn Xuân My (Chủ biên)
- 2. Tin học dùng cho học sinh tiểu học - Quyển I**
Nguyễn Xuân Huy (Chủ biên)
- 3. Nghề tin học ứng dụng** *Bùi Văn Thanh, Ngô Ánh Tuyết*
- 4. Lập trình Pascal** (tập một, tập hai, tập ba)
Bùi Việt Hà
- 5. Tìm đường trong mê cung** *Nguyễn Xuân Huy*

Bạn đọc có thể tìm mua tại các Công ty Sách – Thiết bị trường học ở các địa phương hoặc các Cửa hàng sách của Nhà xuất bản Giáo dục:

Tại Hà Nội : 25 Hàn Thuyên, 81 Trần Hưng Đạo, 187B Giảng Võ, 23 Tràng Tiền.

Tại Đà Nẵng : 15 Nguyễn Chí Thanh.

Tại Thành phố Hồ Chí Minh : 104 Mai Thị Lựu, Quận 1.



Giá: 17.800^d