

TÀI LIỆU LẬP TRÌNH C++ CƠ BẢN

NỘI DUNG:

1/ MẢNG 2 CHIỀU – MA TRẬN

2/ BÀI TẬP

I. MẢNG 2 CHIỀU – MA TRẬN

1. Khai báo mảng 2 chiều

Mảng 2 chiều bản chất là danh sách của các mảng một chiều được xếp liên kế nhau. Việc khai báo mảng hai chiều cũng giống như mảng một chiều, chỉ khác là nó có hai chỉ số đại diện cho số lượng hàng và số lượng cột.

Một mảng 2 chiều có thể coi như là 1 bảng có x hàng và y cột. Một mảng hai chiều a chứa 3 hàng và 4 cột có thể được hiển thị như sau:

	Cột 0	Cột 1	Cột 2	Cột 3
Hàng 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Hàng 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Hàng 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

Khai báo:

<kiểu dữ liệu> <tên biến mảng>[số hàng][số cột];

Ví dụ khai báo mảng a có 100 hàng, 10^5 cột:

```
int a[105][100005];
```

Mảng hai chiều còn gọi là ma trận. Trong ví dụ trên, a là ma trận cấp 3×4 . Để truy cập đến phần tử hàng thứ i, cột thứ j của mảng hai chiều a ta dùng cách viết: a[i][j], chú ý thứ tự hàng trước cột sau.

2. Nhập dữ liệu cho mảng

Để nhập dữ liệu cho một mảng hai chiều, ta phải dùng hai vòng lặp duyệt theo hai chỉ số, chẳng hạn muốn nhập dữ liệu cho mảng a có m dòng, n cột, ta viết:

```
cin>>m>>n;
for (int i=0; i<m; i++)
    for (int j=0; j<n; j++) cin>>a[i][j];
```

3. In các phần tử

```
for (int i=0; i<m; i++)
{
    for (int j=0; j<n; j++) cout<<a[i][j]<<" ";

    cout<<"\n";
}
```

II. BÀI TẬP

1. MA TRẬN

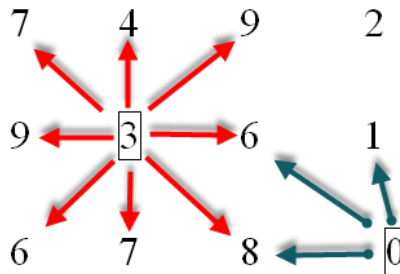
Nhập vào một mảng 2 chiều cấp $m \times n$. Thực hiện cài đặt các chức năng sau:

- A) Tính tổng của mảng.
- B) Đếm số lượng số nguyên tố trong mảng.
- C) Tìm phần tử lớn nhất và các vị trí của phần tử đó trong mảng.
- D) Kiểm tra tính đối xứng của mảng: Mảng 2 chiều được gọi là đối xứng khi $a[i][j]=a[j][i]$.
- E) Sắp xếp lại các hàng của mảng theo kiểu tăng dần từ trái qua phải, giữ nguyên chỉ số hàng.
- F) Xóa bỏ cột đầu tiên của mảng.
- G) Hoán đổi giá trị của dòng đầu tiên và dòng cuối cùng.
- H) Sắp xếp lại mảng theo kiểu tăng dần từ trái qua phải, từ trên xuống dưới.

matran.inp					matran.out
4	5				Tong: 246
4	8	9	12	19	Nguyen to: 12
8	13	17	13	12	Phan tu lon nhat: 19
9	17	12	3	7	Vi tri: a[1,5], a[5,1]
12	13	3	3	1	Mang doi xung
A[i, j]					Mang sap xep:
A[1,1]	a[1,2]	a[1,3]			4 8 9 12 19
A[2,1]					8 12 13 13 17
A[3,1]					3 9 7 12 17
					1 3 3 12 13
					1 7 12 12 19
					Xoa bo cot dau tien:
					8 9 12 19

3. CỰC TIỂU

Một phần tử được gọi là điểm cực tiểu của mảng 2 chiều $A[m,n]$ nếu nó không lớn hơn các phần tử kề nó. Hãy lập trình tìm các điểm cực tiểu của ma trận (cho $m,n \leq 100$, các giá trị của phần tử mảng có giá trị tuyệt đối không quá 32000)



CUCTIEU.INP	CUCTIEU.OUT
3 4	(2, 2)
7 4 9 2	(3, 4)
9 3 6 1	
6 7 8 0	

Chú ý: Đối với phần tử ở biên, ta có 2 cách giải quyết:

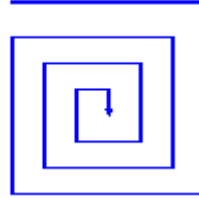
Cách 1: Xử lý riêng trường hợp phần tử nằm ở biên

Cách 2: Mở rộng mảng hai chiều, để chỉ cần một cách so sánh với 8 phần tử xung quanh là vẫn giải quyết được bài toán. Mở rộng mảng nghĩa là bao quanh mảng hai chiều bằng một loạt các phần tử “Max”.

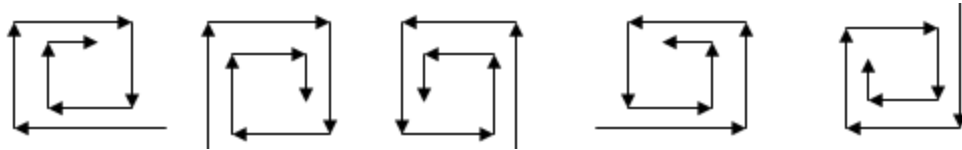
4. XOẮN ỐC

Cho ma trận các số nguyên $A[m,n]$, hãy sắp xếp lại ma trận sao cho các phần tử của chúng có giá trị tăng dần theo đường xoắn ốc từ ngoài vào trong như hình vẽ sau:

1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8
13	12	11	10	9



Có thể sắp xếp theo các dạng như sau:



XOANOC.INP	XOANOC.OUT
3 4	1 2 3 4
7 4 9 2	10 11 12 5
9 3 12 1	9 8 7 6
6 11 8 10	

5. PHẦN TỬ YÊN NGỰA

Cho mảng 2 chiều A có kích thước $M \times N$ số nguyên. Phần tử $A[i][j]$ được gọi là phần tử yên ngựa nếu nó là phần tử nhỏ nhất trong hàng i đồng thời là phần tử lớn nhất trong cột j .

Em hãy lập chương trình tìm phần tử yên ngựa của mảng A .

YENNGUA.INP	YENNGUA.OUT
3 3	(2, 2)
7 3 9	
8 4 6	
5 1 2	

6. Ô CHỮ

Cho một ma trận chữ cái kích thước $m \times n$ và một từ khóa. Viết chương trình in ra các tọa độ chứa các chữ cái tạo nên từ khóa. Các tọa độ phải nằm trên một đường ngang, đường dọc, đường chéo lên hoặc xuống. Giả thuyết trong ma trận chỉ có duy nhất một từ khóa.

Ochu.inp	Ochu.out
WRITE	4 6
8 10	5 7
A W I U W R I E L N	6 8
N R R S R E N N O O	7 9
D E E L L A C D N T	8 10
O I A S N W W O G H	
T L D E O H R W I E	
H N L O P I O I N N	
E A N D I L T N T I	
N D O W P E E T S E	