

## Bài 1. Ma trận đơn vị

Tên file: MTDV.CPP hoặc MTDV.PAS

Viết chương trình nhập vào ma trận nguyên a có n dòng, n cột chỉ chứa hai giá trị 0 hoặc 1.

- Kiểm tra a có phải là ma trận đơn vị không? Nếu có trả về 1, ngược lại trả về 0. Ma trận đơn vị là ma trận vuông mà các phần tử trên đường chéo chính đều bằng 1 còn các phần tử khác đều có giá trị bằng 0.
- Đếm số lần xuất hiện số 0 trong ma trận.

**Input: MTDV.INP**

- Dòng 1 chứa số nguyên dương  $n \leq 100$
- n dòng tiếp theo, mỗi dòng chứa n số nguyên.

**Output: MTDV.OUT**

- Dòng 1, thông báo 1 nếu a là ma trận đơn vị và 0 nếu ngược lại
- Dòng 2, số lượng số 0 trong ma trận a

Ví dụ:

MTDV.INP	MTDV.OUT
3 1 0 0 0 1 0 0 0 1	1 6

## Bài 2. Các tổng

Tên file: MTSUM.CPP hoặc MTSUM.PAS

Cho một ma trận nguyên kích thước  $m \times n$ . Tính:

- Tổng tất cả các phần tử của ma trận.
- Tổng tất cả các phần tử dương của ma trận.
- Tổng tất cả các phần tử âm của ma trận.
- Tổng tất cả các phần tử chẵn của ma trận.
- Tổng tất cả các phần tử lẻ của ma trận.

**Input: MTSUM.INP**

- Dòng 1 chứa hai số nguyên dương m và n ( $m, n \leq 100$ )
- m dòng tiếp theo chứa n số nguyên.

**Output: MTSUM.OUT**

- Gồm 5 dòng, mỗi dòng là kết quả của một câu tương ứng với đề bài.

Ví dụ

MTSUM.INP	MTSUM.OUT
2 3 1 2 3 -4 2 3	7 11 -4 0 7

### Bài 3. Giá trị min max

Tên file: MINMAX.CPP hoặc MINMAX.PAS

Cho một ma trận nguyên kích thước  $m \times n$ . Tìm: Số nhỏ nhất, lớn nhất (kèm chỉ số) của ma trận.

#### Input: MINMAX.INP

- Dòng 1 chứa hai số nguyên dương  $m, n$  ( $m, n \leq 100$ );
- $m$  dòng tiếp theo, mỗi dòng  $i$  chứa  $n$  số nguyên thể hiện các phần tử dòng thứ  $i$  của ma trận

#### Output: MINMAX.OUT

- Dòng 1 chứa 3 số: số thứ nhất là giá trị min của ma trận, 2 số tiếp theo là chỉ số dòng và chỉ số cột của nó (nếu có nhiều giá trị min trong ma trận thì đưa ra chỉ số dòng, cột nhỏ nhất).
- Dòng 2 chứa 3 số: số thứ nhất là giá trị max của ma trận, 2 số tiếp theo là chỉ số dòng và chỉ số cột của nó (nếu có nhiều giá trị max trong ma trận thì đưa ra chỉ số dòng, cột nhỏ nhất).

Ví dụ:

MINMAX.INP	MINMAX.OUT
2 3	-4 2 1
1 2 3	3 1 3
-4 2 3	

### Bài 4. Min max hàng

Tên file: MMROW.CPP hoặc MMROW.PAS

Cho một ma trận nguyên kích thước  $m \times n$ . Tìm: Số nhỏ nhất, lớn nhất (kèm chỉ số) của từng hàng của ma trận.

#### Input: MMROW.INP

- Dòng 1 chứa hai số nguyên dương  $m, n$  ( $m, n \leq 100$ );
- $m$  dòng tiếp theo, mỗi dòng  $i$  chứa  $n$  số nguyên thể hiện các phần tử dòng thứ  $i$  của ma trận

#### Output: MMROW.OUT

- Gồm  $m$  dòng: dòng thứ  $i$  gồm 4 số nguyên:
  - + Số thứ nhất là giá trị nhỏ nhất của dòng  $i$
  - + Số thứ hai là chỉ số cột của giá trị nhỏ nhất tại dòng  $i$
  - + Số thứ ba là giá trị lớn nhất của dòng  $i$
  - + Số thứ tư là chỉ số cột của giá trị lớn nhất tại dòng  $i$

Ví dụ:

MMROW.INP	MMROW.OUT
3 4	-6 4 5 3
-4 2 5 -6	0 3 5 4
3 2 0 5	-2 1 3 2
-2 3 0 2	

## Bài 5. Min max cột

Tên file: MMCOL.CPP hoặc MMCOL.PAS

Cho một ma trận thực kích thước  $m \times n$ . Tìm: Số nhỏ nhất, lớn nhất (kèm chỉ số) của từng cột của ma trận.

**Input: MMCOL.INP:**

- Dòng 1 chứa hai số nguyên dương  $m, n$  ( $m, n \leq 100$ );
- $m$  dòng tiếp theo, mỗi dòng  $i$  chứa  $n$  số nguyên thể hiện các phần tử dòng thứ  $i$  của ma trận

**Output: MMCOL.OUT**

- Gồm  $n$  dòng: dòng thứ  $i$  gồm 4 số nguyên:
  - + Số thứ nhất là giá trị nhỏ nhất của cột  $i$
  - + Số thứ hai là chỉ số dòng của giá trị nhỏ nhất tại cột  $i$
  - + Số thứ ba là giá trị lớn nhất của cột  $i$
  - + Số thứ tư là chỉ số dòng của giá trị lớn nhất tại cột  $i$

Ví dụ:

MINMAX.INP	MINMAX.OUT
3 4	-4 1 3 2
-4 2 5 -6	2 1 3 3
3 2 0 5	0 2 5 1
-2 3 0 2	-6 1 5 2

## Bài 6. Ma trận đối xứng

Tên file: MTDX.CPP hoặc MTDX.PAS

Viết chương trình thực hiện các công việc sau:

- Nhập vào một ma trận  $A$  các giá trị thực kích thước  $n \times n$ .
- Kiểm tra xem ma trận  $A$  có là ma trận đối xứng qua đường chéo chính không?
- Nếu như ma trận là đối xứng qua đường chéo chính thì in ra phần dưới đường chéo chính của ma trận. Nếu không phải ma trận đối xứng thì tạo ra ma trận đối xứng qua đường chéo chính bằng cách sao chép toàn bộ các phần tử dưới đường chéo chính lên các phần tử trên đường chéo chính.

**INPUT: MTDX.INP**

- Dòng 1 chứa số nguyên dương  $n$  ( $n \leq 100$ )
- $N$  dòng tiếp theo, mỗi dòng chứa  $n$  số thực

**OUTPUT: MTDX.OUT**

- Dòng 1: thông báo “YES” nếu  $A$  là ma trận đối xứng, “NO” nếu  $A$  không đối xứng.
- Nếu hiện số 1 ở dòng 1 thì đưa ra nửa ma trận ở các dòng tiếp theo, ngược lại thì đưa ra ma trận đã được tạo đối xứng

Ví dụ:

MTDX.INP	MTDX.OUT	MTDX.INP	MTDX.OUT
4	YES	4	NO
1 5 4 5	1	1 9 7 5	1 5 4 5
5 2 1 3	5 2	5 2 6 3	5 2 1 3
4 1 4 1	4 1 4	4 1 4 5	4 1 4 1
5 3 1 0	5 3 1 0	5 3 1 0	5 3 1 0

## Bài 7. Tìm số nguyên tố

Tên file: MTNT.CPP hoặc MTNT.PAS

Viết chương trình thực hiện các công việc sau:

- Nhập vào ma trận số nguyên A kích thước  $m \times n$
- Kiểm tra xem ma trận A có tồn tại hay không một hàng mà các phần tử đó tăng dần từ trái sang phải.
- Tìm số nguyên tố lớn nhất có trong mảng. Nếu không có thì thông báo -1

**Input: MTNT.INP**

- Dòng 2 chứa số nguyên dương  $m, n$  ( $1 \leq m, n \leq 100$ )
- $m$  dòng tiếp theo, mỗi dòng chứa  $n$  số nguyên

**Output: MTNT.OUT**

- Dòng 1: Nếu tồn tại 1 hàng của ma trận A mà các phần tử của hàng tăng dần từ trái sang phải thì hiện thông báo "YES", ngược lại thì hiện thông báo "NO"
- Dòng 2: Đưa ra giá trị của số nguyên tố lớn nhất trong mảng, ngược lại đưa ra -1.

Ví dụ:

MTNT.INP	MTNT.OUT
3 4 -4 2 5 6 3 2 0 5 -2 3 0 2	YES 5

## Bài 8. Phần tử yên ngựa

Tên file: YENNGUA.CPP hoặc YENNGUA.PAS

Phần tử yên ngựa của một ma trận gọi là phần tử "yên ngựa" nếu như:

- Hoặc phần tử đó lớn nhất trên dòng và nhỏ nhất trên cột
- Hoặc phần tử đó nhỏ nhất trên dòng và lớn nhất trên cột

Viết chương trình nhập vào ma trận  $A[n][m]$ , đưa ra tọa độ của các phần tử yên ngựa có trong ma trận.

**Input: YENNGUA.INP**

- Dòng 1: chứa hai số nguyên  $n, m$  ( $1 \leq n, m \leq 100$ )
- $m$  dòng tiếp theo, mỗi dòng chứa  $n$  số nguyên của ma trận A

**Output: YENNGUA.OUT**

- Đưa ra tọa độ của các phần tử yên ngựa theo trình tự tăng dần theo dòng, rồi đến theo cột.

Ví dụ:

YENNGUA.INP	YENNGUA.OUT
3 4 -4 2 5 5 3 2 8 5 -2 0 7 7	1 3 1 4 2 2

## Bài 9. Tạo mảng B

Tên file: BARR.CPP hoặc BARR.PAS

Viết chương trình nhập vào ma trận  $A[m][n]$

- Hãy xây dựng mảng  $B[m]$  với  $B[i]$  bằng tổng tất cả các phần tử trên dòng  $i$  của ma trận  $A$ .
- Hãy cho biết các dòng có tổng lớn nhất và các dòng có tổng nhỏ nhất trong ma trận  $A$ .

INPUT: BARR.INP

- Dòng 1: chứa hai số nguyên  $n, m$  ( $1 \leq n, m \leq 100$ )
- $m$  dòng tiếp theo, mỗi dòng chứa  $n$  số nguyên của ma trận  $A$

OUTPUT: BARR.OUT

- Dòng 1: chứa  $m$  số nguyên là tổng của các phần tử trên mỗi hàng của ma trận  $A$
- Dòng 2: đưa ra chỉ số dòng có tổng lớn nhất và dòng có tổng nhỏ nhất của ma trận  $A$ .

Ví dụ:

MTNT.INP	MTNT.OUT
3 4	-3 10 3
-4 2 5 -6	2 1
3 2 0 5	
-2 3 0 2	

## Bài 10. Bóng đá

Tên file: FOOTBALL.CPP hoặc FOOTBALL.PAS

Bảng kết quả của một giải vô địch bóng đá được cho bởi ma trận  $A[n*n]$  (có  $n$  đội tham gia và mỗi đội phải đá vòng tròn 1 lượt, tức là mỗi đội phải đá  $n - 1$  trận).

Trong đó:

- + Tất cả các phần tử thuộc đường chéo chính bằng 0
- +  $A[i][j] = 3$  nếu đội  $i$  thắng đội  $j$ , và đội  $i$  có 3 điểm.
- +  $A[i][j] = 1$  nếu đội  $i$  hòa với đội  $j$ , và đội  $i$  có 1 điểm.
- +  $A[i][j] = 0$  nếu đội  $i$  thua đội  $j$ , và đội  $i$  có 0 điểm.

Hãy thực hiện các công việc sau:

- In ra màn hình tất cả các đội có số điểm lớn nhất.
- In ra màn hình tất cả các đội có số trận thắng nhiều hơn thua?
- Hãy chỉ ra các đội không thua trận nào?

Input: FOOTBALL.INP

- Dòng 1 chứa số nguyên dương  $n$  ( $n \leq 50$ );
- $n$  dòng tiếp theo, mỗi dòng chứa  $n$  số nguyên là số điểm của mỗi đội.

Output: FOOTBALL.INP

- Dòng 1 là kết quả câu a
- Dòng 2 là kết quả câu b

- Dòng 2 là kết quả câu c

Ví dụ:

FOOTBALL.INP	FOOTBALL.OUT
4	2
0 0 0 1	2
3 0 3 1	2 4
3 0 0 1	
1 1 1 0	

### Bài 11. Ma trận xoáy ốc

Tên file: XOAYOC.CPP hoặc XOAYOC.PAS

Cho hai số m và n. Hãy tạo một ma trận xoáy ốc từ ngoài vào trong theo chiều kim đồng hồ với kích thước của bảng là  $m \times n$  ( $1 \leq m, n \leq 100$ ).

Ví dụ:  $m = n = 4$  thì ta có ma trận xoáy ốc như sau:

1	2	3	4
12	13	14	5
11	16	15	6
10	9	8	7

**Input: XOAYOC.INP**

- Chứa hai số nguyên dương m và n

**Output: XOAYOC.OUT**

- Đưa ra các dòng của ma trận xoáy ốc

Ví dụ:

XOAYOC.INP	XOAYOC.OUT
3 3	1 2 3 8 9 4 7 6 5

### Bài 12. Ma trận xoáy ốc 2

Tên file: XOAYOC2.CPP hoặc XOAYOC2.PAS

Nhập từ bàn phím một ma trận  $A=(a_{ij})$  kích thước  $m \times n$ . Sau đó sắp xếp mảng A thành một ma trận xoáy ốc từ ngoài vào trong theo chiều kim đồng hồ.

VD: Ma trận kích thước  $4 \times 4$  trước và sau khi sắp xếp:

5	2	5	7
3	5	7	11
4	7	9	10
5	7	12	6

→

2	3	4	5
7	9	10	5
7	12	11	5
7	7	6	5

**Input: XOAYOC2.INP**

- Dòng đầu tiên chứa hai số nguyên dương m và n ( $1 \leq m, n \leq 100$ )
- m dòng tiếp theo, dòng thứ i chứa n số nguyên thể hiện các phần tử của ma trận

**Output: XOAYOC2.OUT**

- Gồm m dòng, mỗi dòng chứa n số nguyên thể hiện ma trận kết quả tìm được

Ví dụ:

XOAYOC2.INP	XOAYOC2.OUT
4 4	2 3 4 5
5 2 5 7	7 9 10 5
3 5 7 11	7 12 11 5
4 7 9 10	7 7 6 5
5 7 12 6	