



Phần 4

Nội dung thực hành

- Vận dụng các cấu trúc điều khiển
- Mảng số 2 chiều: *khai báo, nhập/xuất, thao tác với các phần tử trong mảng*

Ví dụ

1. Nhập dữ liệu cho mảng 2 chiều và in mảng đó ra màn hình

```
#include <stdio.h>

main() {
    int i, j, row, col, a[50][50]; // Khai báo mảng a có 50x50 phần tử

    puts("Nhập hàng(row) và cột(col) của ma trận:");
    scanf("%d%d", &row, &col);

    for(i = 0; i < row; i++) {
        for(j = 0; j < col; j++) {
            printf("Phần tử a[%d][%d]= ", i, j);
            scanf("%d", &a[i][j]); // Nhập giá trị cho các phần tử của mảng
        }
    }

    printf("\nMa trận vừa nhập là:\n");
    for(i = 0; i < row; i++) {
        for(j = 0; j < col; j++)
            printf("%d\t", a[i][j]); // In ra các phần tử của mảng đang hàng - cột

        printf("\n");
    }
}
```

Bài tập

1. Cho một bảng số gồm m hàng và n cột. Tìm giá trị nhỏ nhất và lớn nhất của bảng số đó.

Input

dòng đầu là 2 số m và n

m dòng tiếp, mỗi dòng chứa n số là các phần tử của bảng số

Output

2 số là giá trị nhỏ nhất và lớn nhất của bảng số

Input	Output
3 4	-6 6
2 3 -2 6	
0 0 3 -6	
1 0 2 -3	

2. Cho số n . In ra tam giác Pascal với n dòng.

Input

1 số n duy nhất

Output

n dòng là chứa các hệ số của tam giác Pascal cách nhau bởi dấu cách

Input	Output
5	1 1 1 1 2 1 1 3 3 1 1 4 6 4 1

🔗 : Các dòng của tam giác Pascal chứa các hệ số của khai triển $(a + b)^n$.

3. (Algebra)

Cho một ma trận vuông có $n \times n$ phần tử. Tính tổng các phần tử trên đường chéo chính của ma trận đó.

Input

dòng đầu là 1 số n

n dòng tiếp, mỗi dòng chứa n số là các phần tử của ma trận

Output

1 số duy nhất là tổng các phần tử trên đường chéo chính

Input	Output
3	11
2 3 5	
8 7 3	
1 4 2	

4. (Algebra)

Cho ma trận A có m hàng và n cột, ma trận B có p hàng q cột. Xét xem hai ma trận A và B có thực hiện được phép cộng ma trận không, nếu có thì thực hiện phép toán và in ra kết quả, nếu không thì in ra dòng chữ "INVALID".

Input

dòng đầu là 2 số m và n

m dòng tiếp, mỗi dòng chứa n số là các phần tử ma trận A

dòng tiếp là 2 số p và q

p dòng tiếp, mỗi dòng chứa q số là các phần tử ma trận B

Output

đáp án của bài toán

Input	Output
2 3	2 2 4
1 2 3	4 6 6
4 5 6	
2 3	
1 0 1	
0 1 0	

5. (Algebra)

Cho ma trận A có m hàng và n cột, và ma trận B có p hàng q cột. Xét xem hai ma trận A và B có thực hiện được phép nhân ma trận không, nếu có thì thực hiện phép toán và in ra kết quả, nếu không thì in ra dòng chữ "INVALID".

Input

dòng đầu là 2 số m và n

m dòng tiếp, mỗi dòng chứa n số là các phần tử ma trận A

dòng tiếp là 2 số p và q

p dòng tiếp, mỗi dòng chứa q số là các phần tử ma trận B

Output

đáp án của bài toán

Input	Output
1 2	6 9 12
2 1	
2 3	
1 2 3	
4 5 6	

6. (Algebra)

Hệ phương trình đại số tuyến tính $Ax = b$ thường được biểu diễn bằng một ma trận mở rộng $[A|b]$. Nếu biến đổi ma trận hệ số này về dạng bậc thang rút gọn ([reduced row echelon form](#)), hệ phương trình sẽ được giải. Có 3 phép biến đổi hàng cơ bản, đó là: trao hai hàng, nhân một hàng với một số, và cộng một hàng với một số lần hàng khác.

Định nghĩa các thao tác tương ứng với các phép biến đổi đó là:

- (S, i, j) : trao hai hàng i và j cho nhau (ký tự 'S' cho "swap")
- (M, i, k) : nhân hàng i với một số k (ký tự 'M' cho "multiply")
- (A, i, k, j) : cộng hàng i với k lần hàng j (ký tự 'A' cho "add")

Cho ma trận A có m hàng và n cột. Chọn một 1 trong 3 tác biến đổi có dạng như trên cho ma trận A , và in ra ma trận kết quả.

Input

dòng đầu là 2 số m và n

m dòng tiếp, mỗi dòng chứa n số là các phần tử ma trận A

dòng tiếp là ký tự biểu thị thao tác và các số cần thiết

Output

ma trận A sau phép biến đổi

Input	Output
3 4	1 0 2 -3
2 3 -2 6	0 0 3 -6
0 0 3 -6	2 3 -2 6
1 0 2 -3	
S 1 3	

🔗 Mở rộng bài toán:

- + Cải tiến chương trình bằng cách sử dụng vòng lặp để liên tục nhập thao tác và thực hiện biến đổi ma trận cho đến khi ma trận trở thành dạng rút gọn.
- + Với ma trận ví dụ, các thao tác sau sẽ biến đổi ma trận về dạng rút gọn:
(S, 1, 3)—(S, 2, 3)—(M, 3, 0.33333)—(A, 2, -2, 1)—(A, 2, 6, 3)—(A, 1, -2, 3)—(M, 2, 0.33333)

7. (Algebra)

Cho ma trận A có m hàng và n cột. Thực hiện thuật toán khử Gauss ([Gaussian elimination](#)) (chưa tráo hàng) trên ma trận A và in ra kết quả.

Input

dòng đầu là 2 số m và n

m dòng tiếp, mỗi dòng chứa n số là các phần tử ma trận A

Output

ma trận A sau khi khử Gauss

Input	Output
3 4	2 1 -1 8
2 1 -1 8	0 0.5 0.5 1
-3 -1 2 -11	0 0 -1 1
-2 1 2 -3	

🔑 Gợi ý: Áp dụng các phép biến đổi hàng sơ cấp đã thực hiện ở bài trước.

🔑 Mở rộng bài toán:

- + Cải tiến để chương trình thực hiện được khử Gauss có tráo hàng.
- + Cải tiến tiếp để chương trình thực hiện được thuật toán khử Gauss-Jordan.
- + Xét hạng của ma trận và phân loại nghiệm của hệ phương trình dựa trên định lý Kronecker-Capelli.
- + Tính định thức của ma trận áp dụng thuật toán khử Gauss.
- + Tính ma trận nghịch đảo áp dụng thuật toán khử Gauss-Jordan.
- + Tìm nghiệm của hệ phương trình tuyến tính $Ax = b$ khi biết ma trận hệ số A và vector cột b .

8. Cho một bảng số gồm m hàng và n cột. Cho q truy vấn có dạng (i, j, k, l) , hãy tính tổng các phần tử của “hình chữ nhật con” của bảng số đã cho biết tọa độ góc trên trái là hàng i cột j , và góc tọa độ góc dưới phải là hàng k cột l .

Input

dòng đầu là 3 số m , n , và q

m dòng tiếp, mỗi dòng chứa n số là các phần tử của bảng số

q dòng tiếp, mỗi dòng chứa 4 số i, j, k, l .

Output

q dòng, mỗi dòng là 1 số để trả lời truy vấn tương ứng

Input	Output
3 4 4	-4
2 3 -2 6	3
0 0 3 -6	-3
1 0 2 -3	3
2 2 3 4	
1 3 3 3	
2 1 2 4	
2 3 2 3	

9. Cho một bảng số gồm m hàng và n cột gồm các phần tử là 0 hoặc 1. Hãy tìm và in ra kích thước của hình vuông gồm các ô của bảng thoả mãn các điều kiện sau:

- Hình vuông là đồng nhất: tức là các ô thuộc hình vuông đó phải ghi các số giống nhau (0 hoặc 1)
- Cạnh hình vuông song song với cạnh bảng
- Kích thước hình vuông là lớn nhất có thể

Input

dòng đầu là 2 số m và n

m dòng tiếp, mỗi dòng chứa n số là các phần tử của bảng số

Output

1 số là kích thước của hình vuông lớn nhất

Input	Output
11 13	7
0 0 0 0 0 1 0 0 0 0 0 0 0	
0 0 0 0 1 1 1 0 0 0 0 0 0	
0 0 1 1 1 1 1 1 1 0 0 0 0	
0 0 1 1 1 1 1 1 1 0 0 0 0	
0 1 1 1 1 1 1 1 1 1 0 0 0	
1 1 1 1 1 1 1 1 1 1 1 0 0	
0 1 1 1 1 1 1 1 1 1 0 0 0	
0 0 1 1 1 1 1 1 1 0 0 0 0	
0 0 1 1 1 1 1 1 1 0 0 0 0	
0 0 0 0 1 1 1 0 0 0 0 1 1	
0 0 0 0 0 1 0 0 0 0 0 1 1	

🔑 Gợi ý: Có thể áp dụng thao tác tính tổng các phần tử của “hình chữ nhật con” của bài trước. Ngoài ra việc nhập dữ liệu cho bài toán có thể dễ dàng hơn nếu thực hiện được thao tác đọc từ *file*.

10. Cho một bảng số gồm m hàng và n cột. Một người xuất phát tại ô nào đó của cột 1, cần sang cột n (tại ô nào cũng được).
 Quy tắc đi: Từ ô (i, j) chỉ được quyền sang 1 trong 3 ô $(i - 1, j + 1)$, $(i, j + 1)$, $(i + 1, j + 1)$. Hãy tìm một hành trình đi từ cột 1 sang cột n sao cho tổng các số ghi trên đường đi là lớn nhất.

Input

dòng đầu là 2 số m và n
 m dòng tiếp, mỗi dòng chứa n số là các phần tử của bảng số

Output

1 số là tổng lớn nhất có thể của đường đi

Input	Output
5 7 9 -2 6 2 1 3 4 0 -1 6 7 1 3 3 8 -2 8 2 5 3 2 1 -1 6 2 1 6 1 7 -2 6 2 1 3 7	41

Giải thích test ví dụ:

	1	2	3	4	5	6	7
1	9	-2	6	2	1	3	4
2	0	-1	6	7	1	3	3
3	8	-2	8	2	5	3	2
4	1	-1	6	2	1	6	1
5	7	-2	6	2	1	3	7

11. Ali33 đi lạc vào một cái hang và thấy n mặt hàng có trọng lượng và giá trị khác nhau, nhưng anh ta chỉ mang theo một cái túi có sức chứa về trọng lượng tối đa là M . Nếu bạn là Ali33, bạn sẽ bỏ vào ba lô những món nào và số lượng bao nhiêu để đạt giá trị cao nhất có thể mang đi được.
 Cho n đồ vật, đồ vật thứ i có giá trị p_i và khối lượng w_i .

Input

dòng 1 là 2 số n và M
 dòng 2 là n số w_i
 dòng 3 là n số p_i

Output

tổng giá trị lớn nhất của ba lô

Input	Output
10 67 23 26 20 18 32 27 29 26 30 27 505 352 458 220 354 414 498 545 473 543	1270

🔖 : Bài toán xếp ba lô ([knapsack problem](#)) là một bài toán tối ưu hóa tổ hợp thường xuất hiện trong kinh doanh, toán tổ hợp, lý thuyết về độ phức tạp tính toán, mật mã học, và toán ứng dụng.

12. Theo thần thoại Hy Lạp, rồng Hydra có 7 cái đầu. Mỗi lần Hercules chém đứt 1 đầu thì 2 đầu mới sẽ mọc ra. Giả thiết ban đầu trong 7 đầu có 3 đầu có độc và 4 đầu không độc, và những đầu mọc mới có thể có độc hoặc không (xác suất 50-50). Biết Hercules luôn chém đầu có độc (nếu có), hỏi xác suất để sau đúng 7 lần chém, Hydra chỉ còn lại đầu không độc là bao nhiêu?



- 🔖 Mở rộng bài toán: Xem xét bài toán với giả thiết khác: số lần chém khác, Hercules không phân biệt được 2 loại đầu, tỉ lệ xác suất mọc đầu khác, hoặc số đầu ban đầu lớn hơn.
- 🔖 : Bài toán liên quan đến khái niệm chuỗi Markov ([Markov chain](#)), một quá trình ngẫu nhiên mô tả sự thay đổi của một hệ toán học từ trạng thái này sang trạng thái khác căn cứ trên các quy luật xác suất đã biết.