

## BÀI TẬP THỰC HÀNH CHƯƠNG 1: MẢNG NHIỀU CHIỀU

**Bài 1.** Viết 01 chương trình thực hiện toàn bộ các công việc sau:

- a.** Khai báo một mảng số nguyên gồm 3 hàng và 4 cột.
- b.** Khởi tạo giá trị của mảng vừa nhập lần lượt như sau:

```
8  4  -1  5
2  2  6  9
11 2  5  4
```

- c.** Xuất lại toàn bộ mảng đang lưu trữ để kiểm chứng.

**Bài 2.** Viết chương trình dùng câu lệnh *while* để nhập và xuất giá trị cho một mảng số nguyên gồm 4 hàng và 3 cột.

*Gợi ý: dựa trên tính chất câu lệnh while và for để xây dựng lại phần nhập và xuất mảng 2 chiều.*

**Bài 3.** Viết 01 chương trình thực hiện toàn bộ các công việc sau:

- a.** Cho phép người dùng nhập vào một mảng số nguyên gồm r hàng và c cột (tối đa 10 hàng và 15 cột, nếu nhập sai thì yêu cầu nhập lại cho đến khi đúng mới tiếp tục).

*Gợi ý: dùng vòng lặp do...while để kiểm soát việc nhập giá trị hàng và cột có hợp lệ hay không?*

- b.** Thực hiện việc tính tổng các giá trị lưu trữ trong mảng.
- c.** Tìm giá trị nhỏ nhất, lớn nhất đang lưu trữ trong mảng.

*Gợi ý: dùng 1 giá trị min để chứa giá trị nhỏ nhất trong mảng. Khởi tạo min là giá trị đầu tiên trong mảng, rồi tiến hành so sánh min với các giá trị còn lại để tìm min. Tương tự cho max.*

- d.** Cho biết vị trí của giá trị nhỏ nhất hay lớn nhất ở hàng mấy? cột bao nhiêu (Giả sử người dùng nhập không có giá trị nào trùng nhau).

*Gợi ý: dựa trên giá trị đã tìm được ở câu c, lưu trữ vị trí ứng với giá trị nhỏ nhất hay lớn nhất (có thể lồng ghép trong lúc tìm min hay max) bằng 2 biến hàng và cột. Vì đề toán có*

## Bài tập thực hành Kỹ thuật lập trình

giả sử không có giá trị trùng nhau trong mảng nên chắc chắn rằng trong mảng chỉ có 1 giá trị nhỏ nhất hoặc lớn nhất, việc lưu trữ giá trị nhờ vào giá trị biến đếm ở 2 vòng lặp khi duyệt mảng.

**Bài 4.** Viết 01 chương trình thực hiện toàn bộ các công việc sau:

- a.** Cho phép người dùng nhập vào một mảng số nguyên gồm r hàng và c cột (tối đa 5 hàng và 6 cột, nếu nhập sai thì yêu cầu nhập lại cho đến khi đúng mới tiếp tục).

Gợi ý: dùng vòng lặp do...while để kiểm soát việc nhập giá trị hàng và cột có hợp lệ hay không?

- b.** Cho người dùng nhập vào một vị trí cột/hàng cần tính tổng. Hãy tiến hành tính tổng các giá trị lưu trữ trong cột/hàng mà người dùng yêu cầu. *Lưu ý:* vị trí cột hay hàng phải hợp lệ.

Gợi ý: vị trí cột hay hàng hợp lệ là vị trí từ 0 đến số lượng phần tử tối đa trên cột hay hàng – 1 (hoặc từ 1 đến số lượng phần tử tối đa trên cột hay hàng).

- c.** Cho người dùng nhập vào một giá trị x bất kỳ. Hãy tìm xem x có tồn tại trong mảng hay không? Nếu có thì tồn tại bao nhiêu lần? Tại các vị trí nào?

Gợi ý: cho người dùng nhập giá trị x cần tìm. Duyệt toàn bộ mảng xem có giá trị nào bằng x hay không? Dùng biến đếm để lưu lại số lần tìm được. Việc lưu trữ giá trị nên dựa trên số lần biến đếm để xuất vị trí dòng và cột.

**Bài 5.** Viết chương trình cho nhập vào một ma trận vuông cấp n (chứa các số nguyên). Sau đó tính tổng các giá trị trên đường chéo chính và đường chéo phụ của ma trận vừa nhập.

*Ví dụ:* cho ma trận

$$A = \begin{bmatrix} 1 & 4 & 0 \\ 8 & 15 & 3 \\ 6 & 9 & 2 \end{bmatrix}$$

Thì tổng các giá trị trên đường chéo chính là  $= 1 + 15 + 2 = 18$ , tổng các giá trị trên đường chéo phụ là  $= 6 + 15 + 0 = 21$ .

Gợi ý: các vị trí điều chéo chính đều là  $[0][0]$ ,  $[1][1]$ ,  $[2][2]$ . Tương tự cho đường chéo phụ

## Bài tập thực hành Kỹ thuật lập trình

**Bài 6.** Viết chương trình cho nhập vào 2 ma trận (chứa các số nguyên). Sau đó tính kết quả tổng và tích hai ma trận đó.

**Bài 7.** Viết 01 chương trình thực hiện các yêu cầu sau:

a. Xây dựng các hàm:

- Nhập vào 1 mảng số nguyên gồm  $r$  hàng và  $c$  cột;
- Xuất giá trị 1 mảng số nguyên gồm  $r$  hàng và  $c$  cột.

**b.** Xây dựng hàm trả về giá trị trung bình cộng của các phần tử trong 1 mảng số nguyên gồm  $r$  hàng và  $c$  cột.

**c.** Xây dựng hàm nhận vào 1 mảng số nguyên gồm  $r$  hàng và  $c$  cột; giá trị số nguyên  $x$ . Kiểm tra xem  $x$  có tồn tại trong mảng hay không và trả về kết quả tương ứng.

**d.** Xây dựng hàm nhận vào 1 mảng số nguyên gồm  $r$  hàng và  $c$  cột (giả sử các giá trị nhập không trùng nhau) và 2 giá trị số nguyên  $x$  và  $y$  ( $x$  và  $y$  phải đều tồn tại trong mảng). Tiến hành hoán đổi 2 giá trị  $x$  và  $y$  này. *Lưu ý: sử dụng lại hàm đã xây dựng ở câu c.*

**e.** Xây dựng hàm nhận vào 1 mảng số nguyên gồm  $r$  hàng và  $c$  cột; vị trí cột cần tính tích. Sau đó trả về tích các giá trị số trong cột đã nhận vào.

**f.** Viết hàm main để kiểm tra các hàm đã xây dựng.

**Bài 8.** Xây dựng các hàm:

- Khởi tạo ngẫu nhiên giá trị cho 1 mảng số nguyên gồm  $n$  hàng và  $n$  cột (ma trận vuông cấp  $n$ );
- Xuất giá trị lưu trữ trong 1 mảng số nguyên gồm  $n$  hàng và  $n$  cột.
- Viết hàm tính tổng, tích 2 ma trận vuông cấp  $n$ .
- Viết hàm main để kiểm tra các hàm đã xây dựng.

## Bài tập thực hành Kỹ thuật lập trình

**Bài 9.** Viết chương trình gán các giá trị số nguyên từ 1 đến  $n^2$  cho các phần tử của ma trận vuông có kích thước  $n \times n$  ( $2 \leq n \leq 10$ ) được nhập từ bàn phím theo dạng zigzag cột. Ví dụ  $n = 4$ :

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

**Bài 10.** Viết chương trình gán các giá trị số nguyên từ 1 đến  $n^2$  cho các phần tử của ma trận vuông có kích thước  $n \times n$  ( $2 \leq n \leq 10$ ) được nhập từ bàn phím theo dạng xoắn ốc. Ví dụ  $n = 4$ :

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

**Bài 11.** Viết chương trình đặt chỗ ngồi cho 1 máy bay. Máy bay có 13 hàng ghế, mỗi hàng có 6 ghế. Hàng 1 và 2 là hạng thương gia, hàng 3 đến 7 là hạng phổ thông, hàng 8 trở đi là hạng tiết kiệm.

Chương trình in ra bảng đặt chỗ ngồi có dạng như sau:

## Bài tập thực hành Kỹ thuật lập trình

		A	B	C	D	E	F
Hang	1	*	*	X	*	X	X
Hang	2	*	X	*	X	*	X
Hang	3	*	*	X	X	*	X
Hang	4	X	X	X	*	X	X
Hang	5	*	X	*	X	*	*
Hang	6	*	X	*	*	*	X
Hang	7	X	*	*	*	X	X
Hang	8	*	X	*	X	X	*
Hang	9	X	*	X	X	*	*
Hang	10	*	X	*	X	X	X
Hang	11	*	*	X	*	X	*
Hang	12	*	*	X	X	*	X
Hang	13	*	*	*	*	X	*

(\*: chỗ còn trống, x: chỗ đã đặt.)

Chương trình yêu cầu người dùng nhập vào loại vé (thương gia, phổ thông hay tiết kiệm) và chỗ ngồi muốn đặt. Chương trình kiểm tra và in thông báo thích hợp, đồng thời có menu cho phép người dùng lựa chọn.

-----**HẾT**-----