

MẢNG 2 CHIỀU - CÚ PHÁP

Các tên hợp lệ của mảng

<Kiểu dữ liệu> tên_mảng [<số dòng>] [<số cột>]

int, float, double, char, object, ...



CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

hoặc <Kiểu dữ liệu> [<số dòng>][<số cột>] tên_mảng

<Kiểu dữ liệu> tên_mảng [<row>] [<column>]

MẢNG 2 CHIỀU - CÚ PHÁP

`arr[3][3]`

col 0 col 1 col 2

row 0

0 0	0 1	0 2
-----	-----	-----

row 1

1 0	1 1	1 2
-----	-----	-----

row 2

2 0	2 1	2 2
-----	-----	-----

MẢNG 2 CHIỀU - MỘT SỐ GHI CHÚ

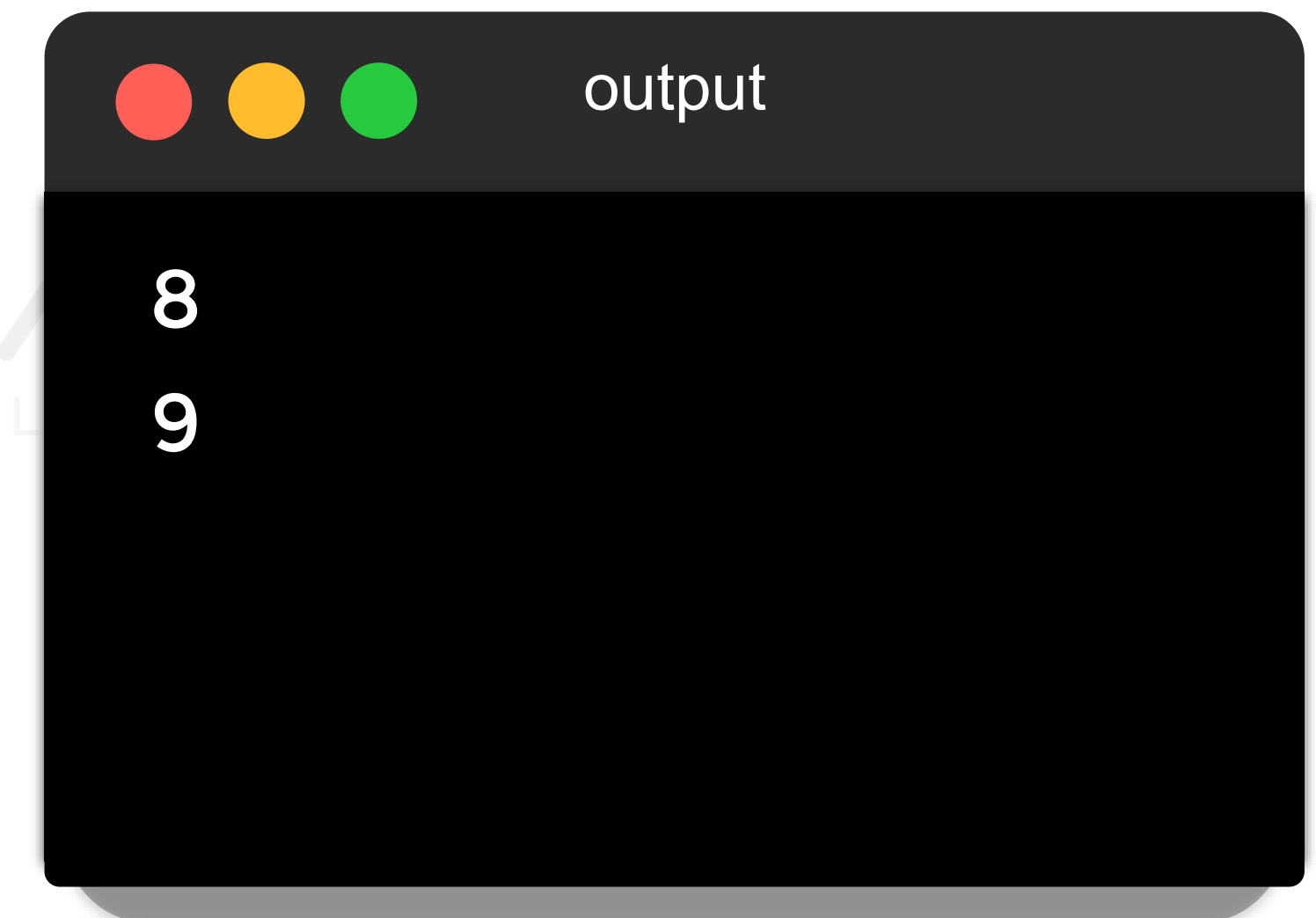
- Số dòng: là một hằng số nguyên, cho biết số lượng dòng tối đa
- ☑ Số cột: là một hằng số nguyên, cho biết số lượng cột tối đa
- ☑ Số phần tử: số dòng * số cột
- ☑



MẢNG 2 CHIỀU - KHỞI TẠO MẢNG

```
int arr[3][3] = {{1, 2, 3} , {5, 7, 8}, {9,8,1}}
```

	col 0	col 1	col 2
row 0	<small>arr[0][0]</small> 1	<small>arr[0][1]</small> 2	<small>arr[0][2]</small> 3
row 1	<small>arr[1][0]</small> 5	<small>arr[1][1]</small> 7	<small>arr[1][2]</small> 8
row 2	<small>arr[2][0]</small> 9	<small>arr[2][1]</small> 8	<small>arr[2][2]</small> 1



```
System.out.print (a[1][2]);
```

```
System.out.print (a[2][0]);
```

MẢNG 2 CHIỀU - HÀM NHẬP

```
void nhapMaTran(int a[][],int soDong, int soCot, Scanner scan) {  
    for (int i = 0; i < soDong; i++) {  
        for (int j = 0; j < soCot; j++) {  
            System.out.print("Nhập phần tử thứ [" + i + ", " + j + "]: ");  
            a[i][j] = Integer.parseInt(scan.nextLine());  
        }  
    }  
}
```

3	7	1
8	-3	-9
1	-7	3

MẢNG 2 CHIỀU - HÀM XUẤT

```
void xuatMaTran(int a[][],int soDong, int soCot) {  
    for (int i = 0; i < soDong; i++) {  
        for (int j = 0; j < soCot; j++) {  
            System.out.print(a[i][j] + "\\t");  
        }  
        // sau khi viết xong 1 dòng thì xuống hàng  
        System.out.println("\\n");  
    }  
}
```

MẢNG 2 CHIỀU - BÀI TẬP 1

Viết chương trình nhập vào mảng 2 chiều, kiểm tra dòng, cột > 1 .
Tính tổng các phần tử trong mảng 2 chiều đó.



MẢNG 2 CHIỀU - BÀI TẬP 2

Tìm max trong ma trận. Các phần tử được tạo ngẫu nhiên trong đoạn $[-50, 50]$



MẢNG 2 CHIỀU - BÀI TẬP 3

Liệt kê các dòng có chứa giá trị chẵn trong ma trận các số nguyên. Các phần tử được tạo ngẫu nhiên trong đoạn $[-50, 50]$

GHI CHÚ:

- * Kỹ thuật đặt cờ hiệu
- * Duyệt theo dòng và đặt cờ FALSE
- * Duyệt theo cột và kiểm tra thỏa điều kiện (Chỉ cần 1 điều kiện đúng là được, đảo cờ TRUE và break

3	7	6	2
2	1	1	5
1	9	7	5



MẢNG 2 CHIỀU - BÀI TẬP 4

Liệt kê các dòng giảm dần trong ma trận. Các phần tử được nhập vào từ bàn phím

GHI CHÚ:

- * Kỹ thuật đặt cờ hiệu
- * Duyệt theo dòng và đặt cờ TRUE (Yêu cầu tất cả thì đặt cờ true
- * Duyệt theo cột và kiểm tra KHÔNG thỏa điều kiện (tìm 1 trường hợp vi phạm), đảo cờ và break

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



MẢNG 2 CHIỀU - BÀI TẬP 5

Viết chương trình thực hiện các công việc sau:

- Nhập m, n là số dòng và số cột của ma trận 2 chiều A và B từ bàn phím.
- Tạo giá trị ngẫu nhiên cho các phần tử trong đoạn $[-10,10]$
- Tính và in ra màn hình ma trận C là tổng của 2 ma trận này.

GHI CHÚ:

- * Hai ma trận cùng số dòng và số cột
- * Giá trị của phần tử trong ma trận C bằng tổng 2 giá trị tại vị trí tương ứng của 2 ma trận A, B

3	7	6	1	3	2	4	10	8
3	2	1	-2	1	0	1	3	1
1	9	7	-1	1	7	0	10	14
A			B			C		

MA TRẬN VUÔNG

	col 0	col 1	col 2
row 0	arr[0][0] 1	arr[0][1] 2	arr[0][2] 3
row 1	arr[1][0] 5	arr[1][1] 7	arr[1][2] 8
row 2	arr[2][0] 9	arr[2][1] 8	arr[2][2] 1

- Mảng 2 chiều có : **$S_{\text{dòng}} = S_{\text{cột}}$**

- Đường chéo chính của ma trận vuông
 $\Rightarrow \text{ch}_{\text{s dòng}} = \text{ch}_{\text{s cột}}$

- Đường chéo phụ của ma trận vuông
 $\Rightarrow \text{ch}_{\text{s dòng}} + \text{ch}_{\text{s cột}} = \text{kích thước} - 1$

NHẬP - XUẤT MA TRẬN VUÔNG

```
int[][] nhapMaTran(int n, Scanner scan) {
    int a[][] = new int[n][n];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            System.out.print("Nhập phần tử thứ [" + i + ", " + j + "]: ");
            a[i][j] = Integer.parseInt(scan.nextLine());
        }
    }
}

void xuatMaTran(int a[][],int n) {
    return a;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            System.out.print(a[i][j] + "\t");
        }
        // sau khi viết xong 1 dòng thì xuống hàng
        System.out.println("\n");
    }
}
```

3	7	1
8	-3	-9
1	-7	3

XUẤT PHẦN TỬ ĐƯỜNG CHÉO CHÍNH MA TRẬN VUÔNG

```
void xuatMaTran(int a[][],int n) {  
    for (int i = 0; i < n; i++) {  
        System.out.print(a[i][i]+"\t");  
    }  
}
```

3	7	1
8	-3	-9
1	-7	3

TỔNG PHẦN TỬ ĐƯỜNG CHÉO PHỤ MA TRẬN VUÔNG

```
int tongCheoPhu(int a[][],int n) {  
    int tong =0;  
    for (int i = 0; i < n; i++) {  
        tong += a[i][n-1-i];  
    }  
    return tong;  
}
```

3	7	1
---	---	---

8	-3	-9
---	----	----

1	-7	3
---	----	---

MA TRẬN VUÔNG - BÀI TẬP 1

Kiểm tra một ma trận có phải là ma trận đối xứng hay không.

GHI CHÚ:

- * Phải là ma trận vuông
- * Phần tử nằm trên và dưới đối xứng qua đường chéo chính phải bằng nhau ($a[i][j] == a[j][i]$)
- * Sử dụng return false nếu vi phạm 1 lần vì kiểm tra tất cả

3	1	4
---	---	---

1	-3	-7
---	----	----

4	-7	3
---	----	---

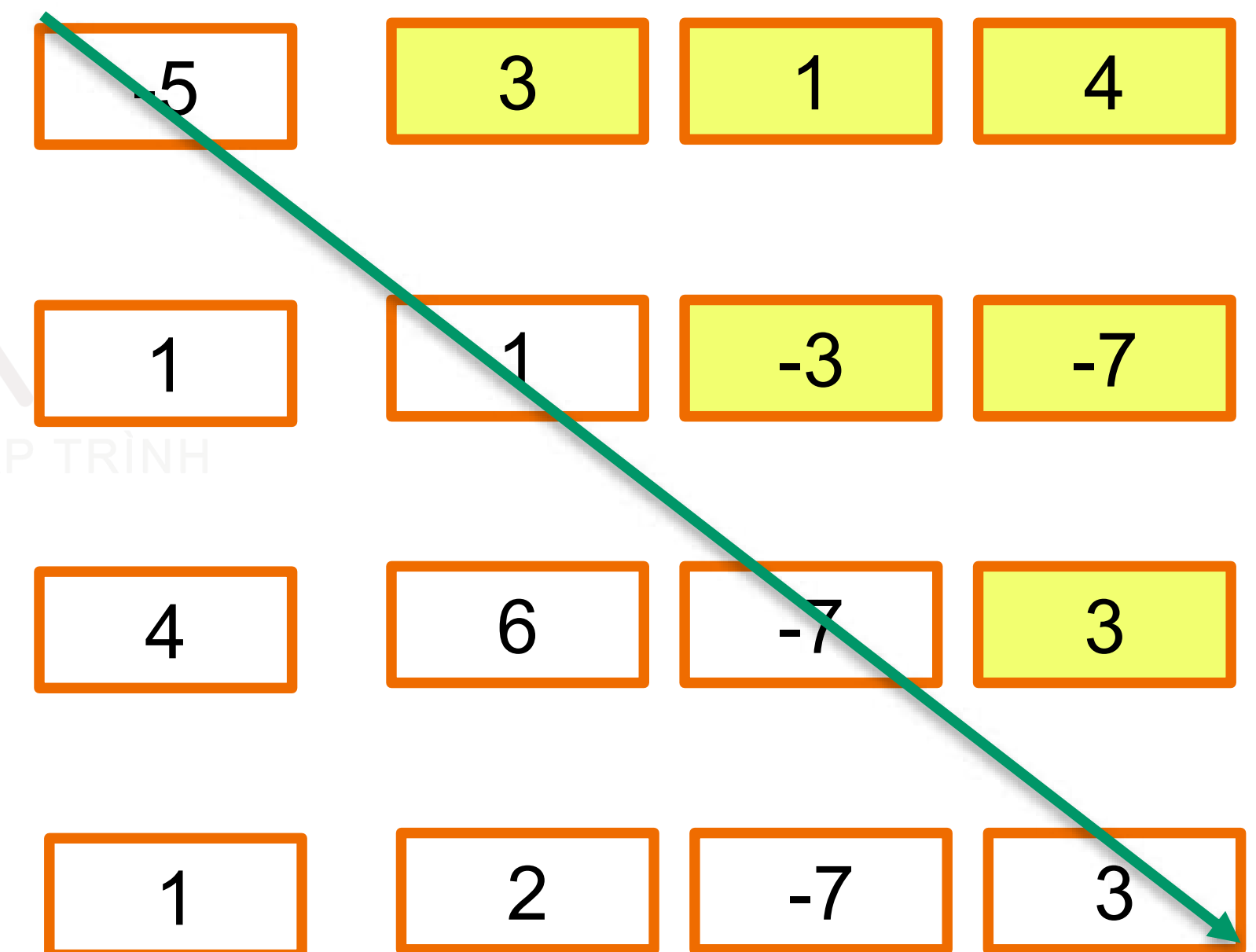
MA TRẬN VUÔNG - BÀI TẬP 2

Tính tổng các phần tử thuộc tam giác trên các đường chéo. Không tính đường chéo

GHI CHÚ:

* Tam giác các phần tử trên của đường chéo chính sẽ có chỉ số là :

- i sẽ chạy từ $0 \rightarrow n-1$
- j sẽ chạy từ $i+1 \rightarrow n$



-5	3	1	4
1	1	-3	-7
4	6	-7	3
1	2	-7	3

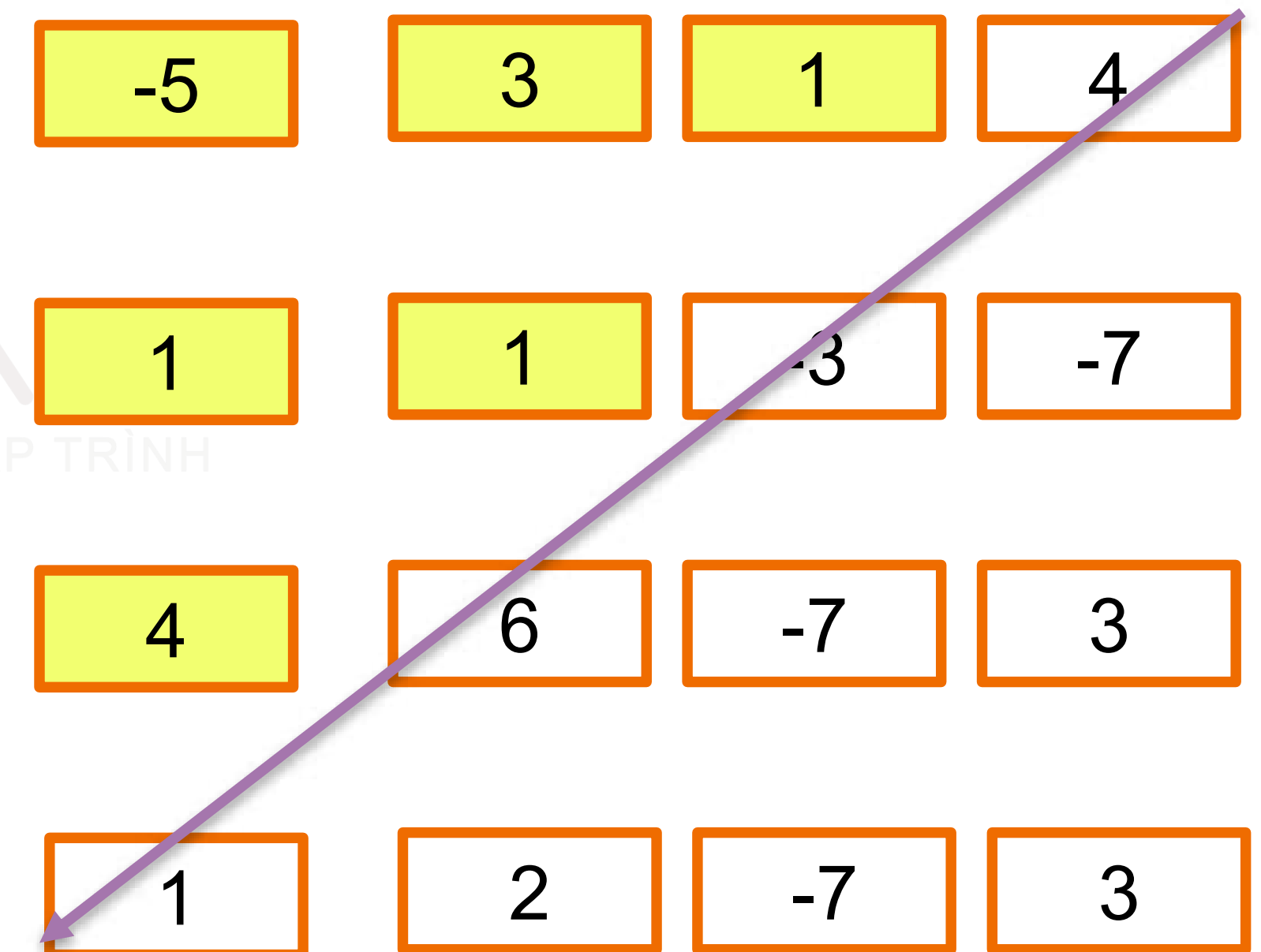
MA TRẬN VUÔNG - BÀI TẬP 2

Tính tổng các phần tử thuộc tam giác trên các đường chéo. Không tính đường chéo

GHI CHÚ:

* Tam giác các phần tử trên của đường chéo chính sẽ có chỉ số là :

- i sẽ chạy từ $0 \rightarrow n-1$
- j sẽ chạy từ $0 \rightarrow n-1-i$



-5	3	1	4
1	1	-3	-7
4	6	-7	3
1	2	-7	3

BÀI TẬP NỘP

1. Tìm số chẵn đầu tiên trong ma trận 2 chiều. Mỗi phần tử được nhập ngẫu nhiên trong đoạn $[-50,50]$
2. Tính tổng các phần tử chẵn trong mảng 2 chiều các số nguyên. Mỗi phần tử được nhập ngẫu nhiên trong đoạn $[-50,50]$
3. Tìm số dương đầu tiên trong ma trận. Mỗi phần tử được nhập ngẫu nhiên trong đoạn $[-50,50]$
4. Tìm giá trị âm lớn nhất trong ma trận. Mỗi phần tử được nhập ngẫu nhiên trong đoạn $[-50,50]$
5. Tìm số nguyên tố đầu tiên. Mỗi phần tử được nhập ngẫu nhiên trong đoạn $[-50,50]$
6. Kiểm tra ma trận có toàn dương hay không. Các phần tử được nhập từ bàn phím.
7. Kiểm tra một cột ma trận có giảm dần hay không. Các phần tử được nhập từ bàn phím.
8. Tính tổng các giá trị trên 1 dòng của ma trận các phần tử được tạo ngẫu nhiên trong đoạn $[-50, 50]$
9. Tính tổng các giá trị lẻ trên 1 cột của ma trận các phần tử được tạo ngẫu nhiên trong đoạn $[-50, 50]$
10. Liệt kê các dòng có chứa giá trị âm trong ma trận các số được tạo ngẫu nhiên trong đoạn $[-50,50]$

BÀI TẬP NỘP

11. Liệt kê các cột tăng dần trong ma trận. Các phần tử nhập vào từ bàn phím
12. Tìm phần tử chẵn cuối cùng các số được tạo ngẫu nhiên trong đoạn $[-50,50]$
13. Tính tổng các phần tử nguyên tố có trong mảng các số được tạo ngẫu nhiên trong đoạn $[-50,50]$
14. Tìm giá trị xuất hiện nhiều nhất các số được tạo ngẫu nhiên trong đoạn $[-50,50]$
15. Một ma trận được gọi là ma trận thừa nếu số phần tử có giá trị bằng 0 nhiều hơn số phần tử khác 0. Viết chương trình thực hiện các công việc sau:
 - Nhập m, n là số dòng và số cột của ma trận hai chiều A từ bàn phím.
 - Nhập giá trị các phần tử của ma trận A từ bàn phím.
 - Kiểm tra và thông báo lên màn hình ma trận vừa nhập là ma trận thừa hay ma trận không thừa.
16. Cho ma trận vuông bậc n (n nhập từ bàn phím) với các phần tử là các trị ngẫu nhiên trong đoạn $[-100, 100]$,Viết hàm xuất tất cả các đường chéo song song với đường chéo phụ, mỗi đường chéo trên một dòng

```
Nhap bac ma tran: 3 ↵
-84  50  68
 53 -94 -47
-62 -59  67
Cac duong cheo song song cheo phu:
-84
 50  53
 68 -94 -62
-47 -59
 67
```