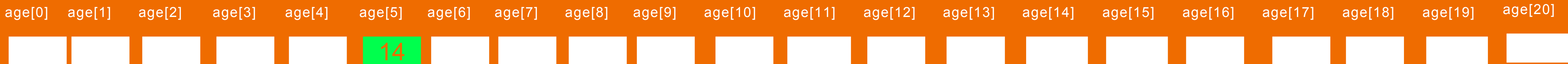


CÁC PHẦN LIÊN QUAN ĐẾN MẢNG

```
int age[20];
```



3. Chỉ số đầu tiên / index

1. Chỉ số của mảng

2. Độ dài của mảng = 20

4. Chỉ số cuối cùng = độ dài - 1

5. Phần tử thứ 6 có giá trị = 14

KHAI BÁO MẢNG

Các tên hợp lệ của mảng

<Kiểu dữ liệu> tên_mảng [] hoặc <Kiểu dữ liệu> [] tên_mảng

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

CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

CẤP PHÁT VÙNG NHỚ

```
// Khai báo và cấp phát vùng nhớ cho mảng kiểu int  
int[] arr1 = new int [5] //a.length = 5;
```

```
// Khai báo và cấp phát vùng nhớ cho mảng kiểu int  
long[] arr2 = new long [8]; //a.length = 8;
```

```
// Khai báo và cấp phát vùng nhớ cho mảng kiểu int  
float[] arr3 = new float [9]; //a.length = 9;
```

```
// Khai báo và cấp phát vùng nhớ cho mảng kiểu int  
String[] arr4 = new String[4]; //a.length = 4;
```

```
// Khai báo và cấp phát vùng nhớ cho mảng kiểu int  
double[] arr5 = new double [6] //a.length = 6;
```

```
// Khai báo và cấp phát vùng nhớ cho mảng kiểu int  
boolean[] arr6 = new boolean [7];
```



CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

KHỞI TẠO MẢNG

// Khởi tạo mảng cách 1

```
int[] arr1 = {3, 4, 1, 5, 6};
```

// Khởi tạo mảng cách 2

```
double[] arr2 = new double [4];
```

```
arr2[0] = 3.4;
```

```
arr2[1] = 6;
```

```
arr2[2] = 5.2;
```

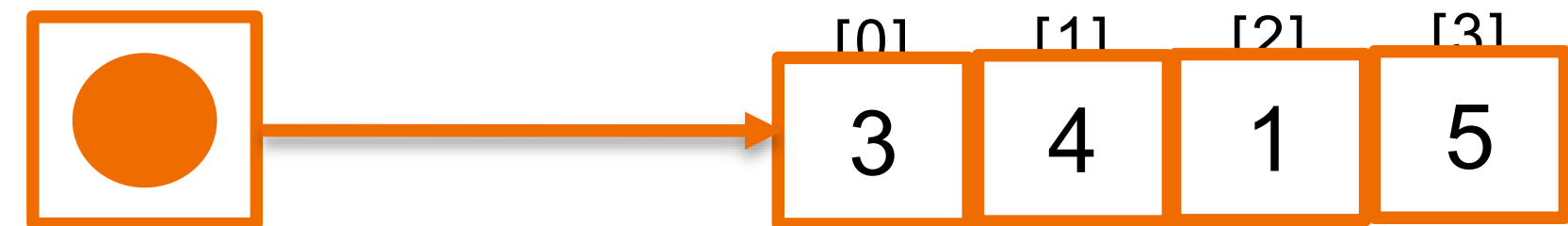
```
arr2[3] = 1.4;
```



CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

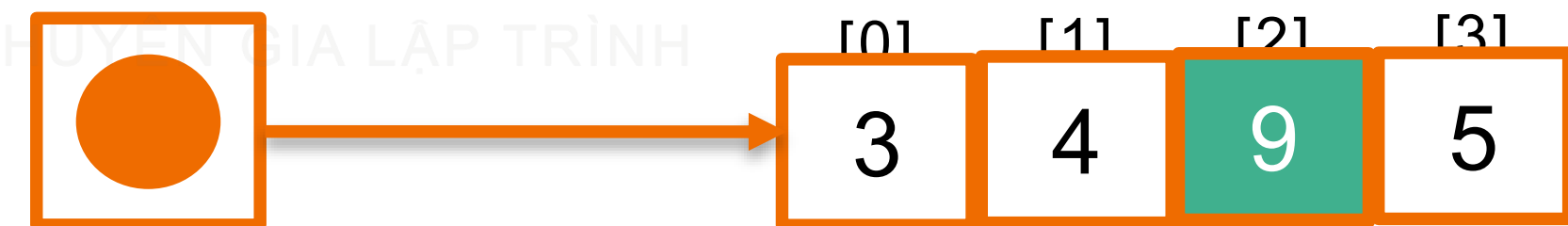
BẢNG CHẤT LƯU TRỮ CỦA MẢNG

numStudent



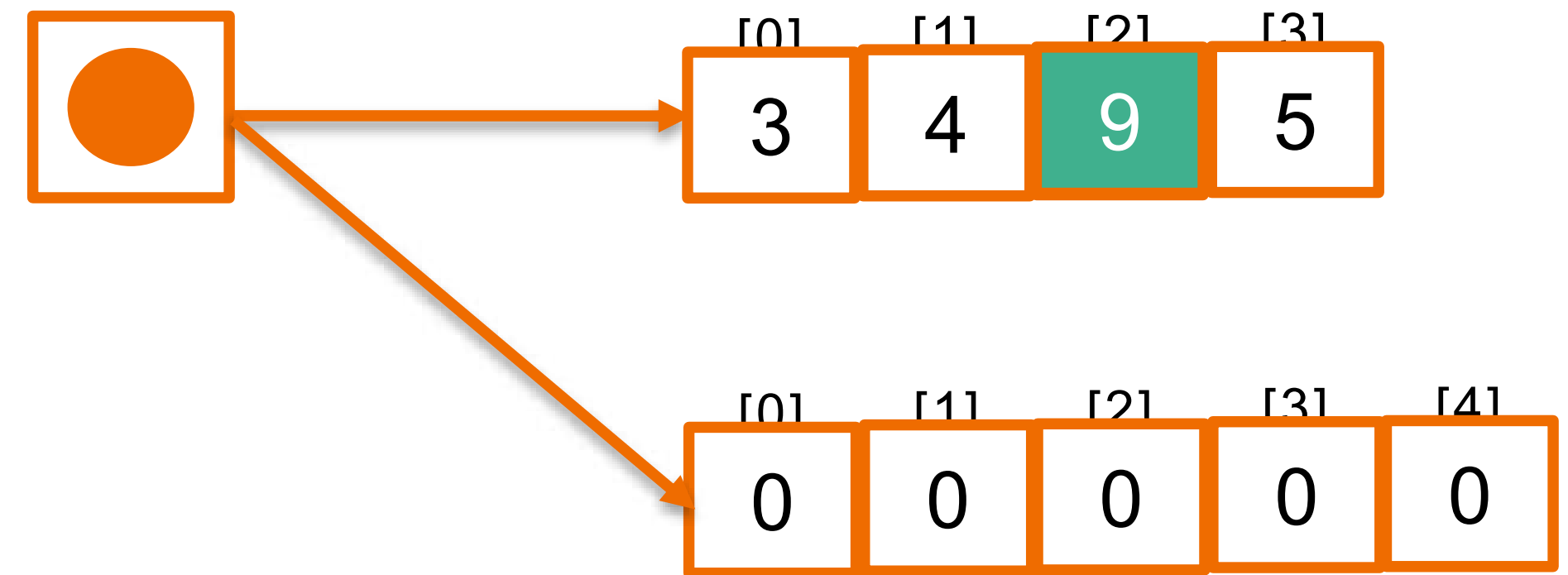
```
int[] numStudent = {3, 4, 1, 5};
```

numStudent



```
numStudent [2] = 9;
```

numStudent



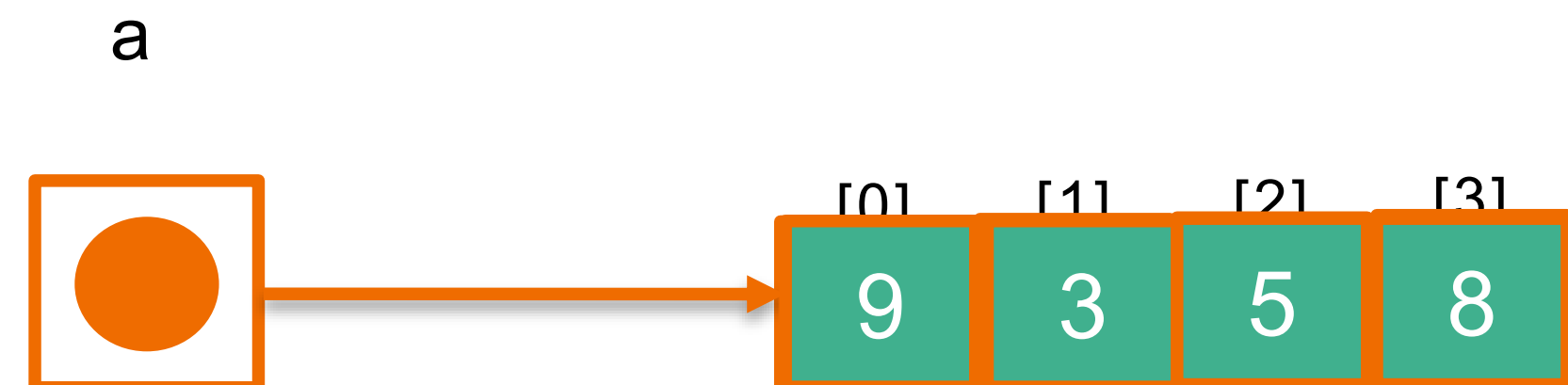
```
numStudent = new int[5];
```

NHẬP MẢNG MỘT CHIỀU

// Nhập mảng

```
System.out.print("Nhập số phần tử của  
mảng là ");  
int n = Integer.parseInt(scan.nextLine());  
int[] a = new int[n];  
for (int i = 0; i < a.length; i++){  
    System.out.print ("a[" + i + "  
    a[i] = Integer.parseInt  
  
}
```

```
output  
  
> Nhập số phần tử của mảng là 4  
a[0] = 9  
a[1] = 3  
a[2] = 5  
a[3] = 8
```



XUẤT MẢNG MỘT CHIỀU DÙNG FOR...I

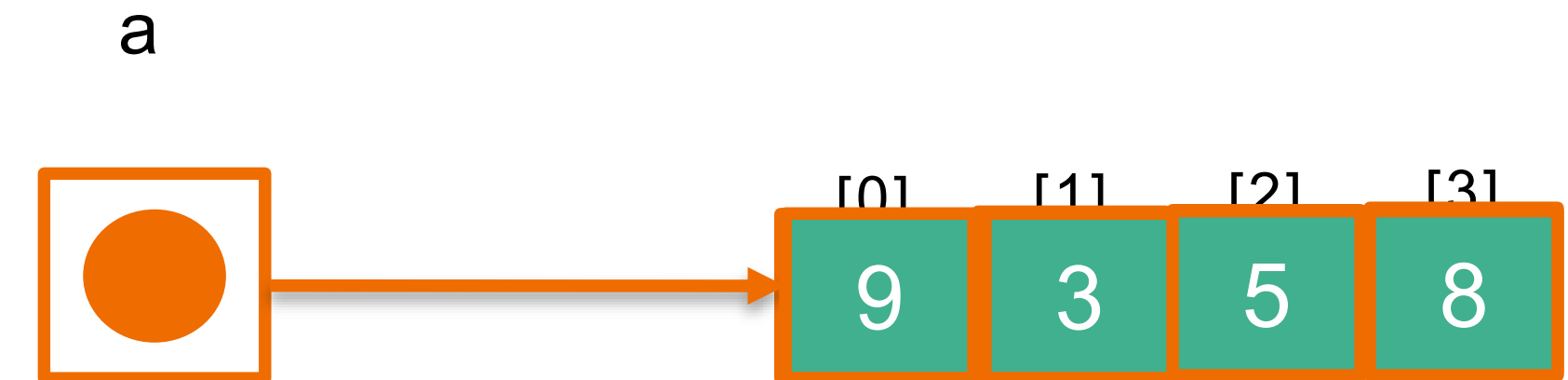
// Xuất mảng

```
System.out.print("Xuất mảng một chiều ");
```

```
System.out.print("Số phần tử của mảng  
là: n = " + a.length);
```

```
for (int i = 0; i < a.length; i++){
```

```
}
```



```
output
Xuất mảng một chiều
Số phần tử của mảng là: n = 4
9
3
5
8
```

XUẤT MẢNG MỘT CHIỀU DÙNG FOR...EACH

```
// Xuất mảng dùng for...each
// tenBien lần lượt là các phần tử bên trong mảng (không duyệt qua index mà duyệt luôn phần tử)
for (KieuDuLieu tenBien : tenMang) {
    //Các câu lệnh
}
System.out.print("Xuất phần tử dùng for...each ");
System.out.print("Số phần tử của mảng là: n = " + a.length);
for (int pt: tenMang){
    System.out.println( pt);
}
```



CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

BÀI TẬP 1: LUYỆN TẬP MẢNG

1. Cho người dùng nhập vào n số nguyên và lưu trữ trong mảng
 - a. Viết hàm nhập, hàm xuất
 - b. Tính tổng các số chẵn (dùng for và for each)
 - c. Đếm có bao nhiêu số âm (dùng for và for each)
 - d. Tổng các số âm (dùng for và for each)



CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

BÀI TẬP 2: LUYỆN TẬP MẢNG

Tạo ngẫu nhiên mảng một chiều n phần tử nguyên (n chẵn) có giá trị chứa trong đoạn $[-100, 100]$ và xuất mảng. Hãy liệt kê các số trong mảng 1 chiều các số nguyên thuộc đoạn $[x, y]$ cho trước

* Sinh số ngẫu nhiên trong đoạn $[a, b]$: $a + (\text{int})(\text{Math.random()} * ((b - a) + 1));$

N 10 = 10 87 -4 9 8 19 -9 -23 89 92
-14 -> 30 -> 10, -4, 9, 8, 19, -9



BÀI TẬP 3: LUYỆN TẬP MẢNG

Tạo ngẫu nhiên mảng một chiều n phần tử - n do người dùng nhập vào, có giá trị chứa trong đoạn $[-100, 100]$ và xuất mảng. Hãy liệt kê các vị trí trong mảng 1 chiều các số thực mà giá trị tại đó bằng giá trị âm đầu tiên trong mảng

Ví dụ:

```
9 8 33 -5 -9 18 2 8 -5 8 33 12 -5 8
19 18 33 5 9 18 12 28 5 18 33 12 55 98
```

Ghi chú & Giải thuật

- 1) Mảng toàn số dương
- 2) Tìm chỉ số của số âm đầu tiên
- 3) Nếu tìm thấy, duyệt lại vòng lặp và so sánh phần tử đang duyệt với giá trị phần tử tại chỉ số âm vừa tìm thấy, in ra chỉ số



BÀI TẬP 4: LUYỆN TẬP MẢNG

Tạo ngẫu nhiên mảng một chiều n phần tử - n do người dùng nhập vào, có giá trị chứa trong đoạn $[-100, 100]$ và xuất mảng. Hãy liệt kê các vị trí mà giá trị tại các vị trí đó bằng giá trị dương nhỏ nhất đầu tiên trong mảng 1 chiều các số thực

Ghi chú & Giải thuật

- 1) Mảng toàn số âm
- 2) Tìm số dương đầu tiên
- 3) Tìm số dương nhỏ nhất thật sự
 - Gán chỉ số **indexMin** là chỉ số của số dương đầu tiên là nhỏ nhất
 - Duyệt mảng, xem số dương nào nhỏ hơn, cập nhật lại chỉ số **indexMin**
 - Tìm thấy **indexMin**
4. Duyệt lại mảng từ vị trí **indexMin**, xem phần tử nào có giá trị bằng với giá trị của phần tử tại **indexMin** => Xuất ra

-4 -12 -9 28 9 -4 6 2 19 -4 2 90 2

BÀI TẬP 5: LUYỆN TẬP MẢNG

Tạo ngẫu nhiên mảng một chiều n phần tử - n do người dùng nhập vào, có giá trị chứa trong đoạn $[-1000, 1000]$ và xuất mảng. Hãy liệt kê các giá trị trong mảng 1 chiều các số nguyên có chữ số đầu tiên là chữ số lẻ

Ghi chú & Giải thuật

- 1) Viết hàm kiểm tra một số có chữ số đầu tiên là số lẻ -> Tách các chữ số trong 1 số
- 2) Duyệt vòng lặp và kiểm tra từng phần tử thỏa mãn yêu cầu là chữ số đầu tiên là số lẻ, in ra phần tử đó

BÀI TẬP 6: LUYỆN TẬP MẢNG

Tạo ngẫu nhiên mảng một chiều n phần tử - n do người dùng nhập vào, có giá trị chứa trong đoạn $[-1000, 1000]$ và xuất mảng.
Tính tổng các giá trị lớn hơn giá trị đứng liền trước nó trong mảng 1 chiều các số thực

Ghi chú & Giải thuật

Liền trước có nghĩa là xét vị trí $a[i]$ và $a[i-1]$, nếu $a[i] > a[i-1]$ thì tích lũy tổng

3 4 7 1 9 8 2 6 4

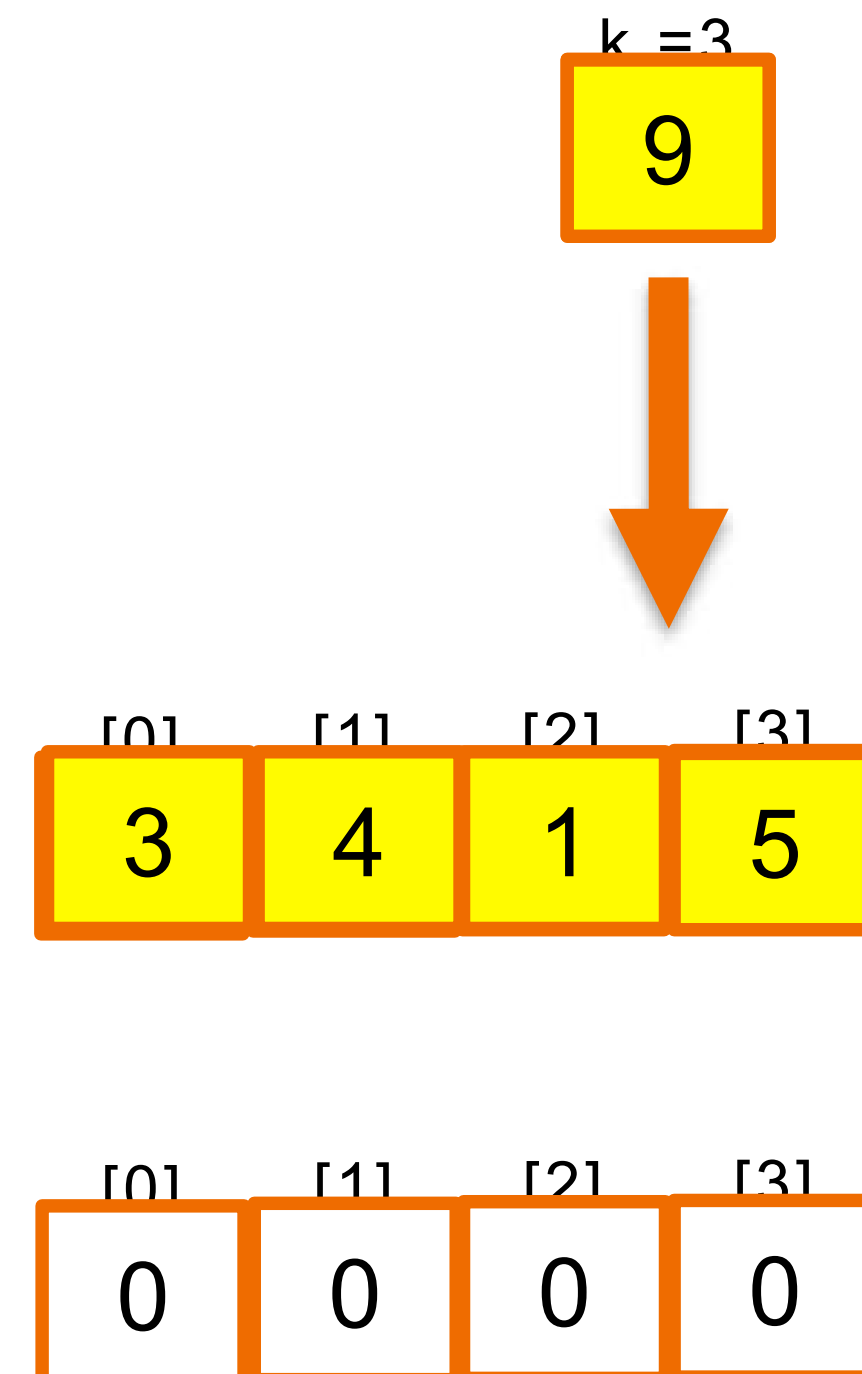


BÀI TẬP 7: LUYỆN TẬP MẢNG

Tạo ngẫu nhiên mảng một chiều n phần tử - n do người dùng nhập vào, có giá trị chứa trong đoạn $[-100, 100]$ và xuất mảng. Thêm 1 phần tử x tại vị trí k do người dùng nhập vào

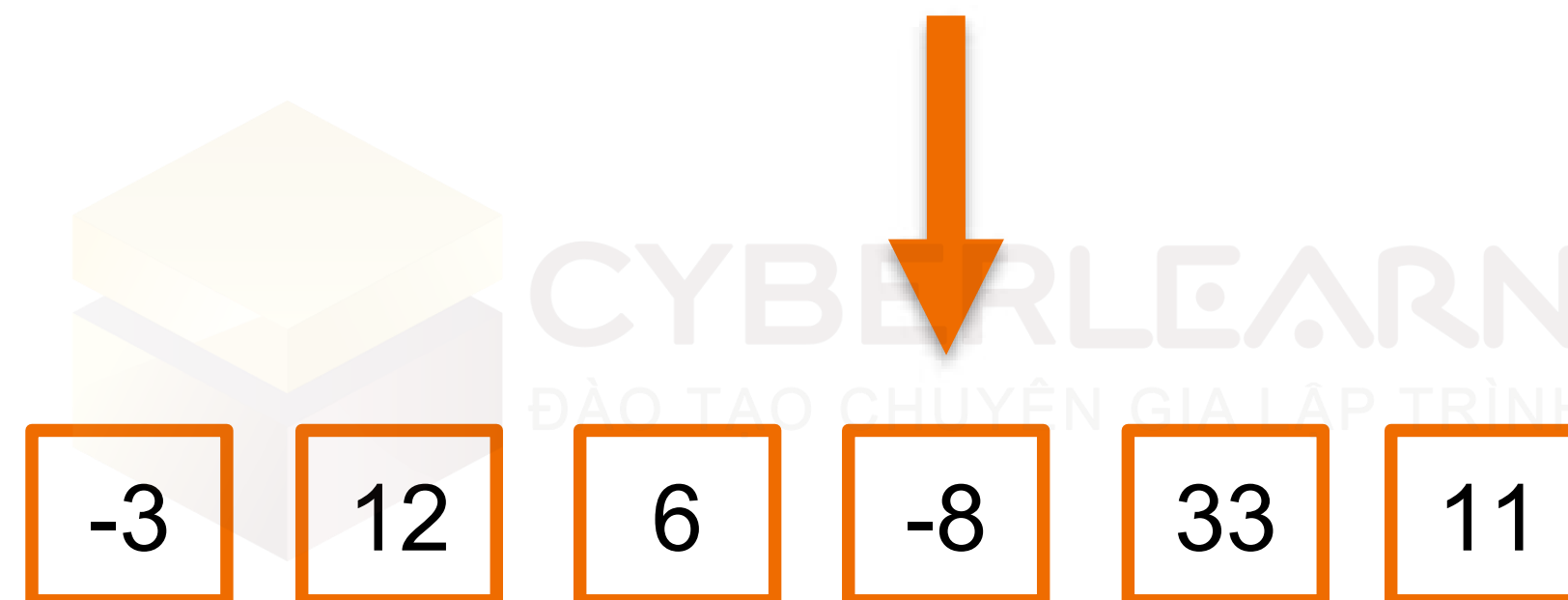
Ghi chú & Giải thuật

Sử dụng kỹ thuật tạo mảng tạm để lưu trữ tạm và sau đó gán lại cho mảng chính. Chú ý về các chỉ số và kích thước của các mảng trước và sau thêm



BÀI TẬP 9X - MẢNG

Tạo ngẫu nhiên mảng một chiều n phần tử nguyên có giá trị chứa trong đoạn $[-100, 100]$ và xuất mảng. Xóa phần tử có chỉ số p (p nhập từ bàn phím) trong mảng. Yêu cầu kiểm tra chỉ số nhập vào hợp lệ



BÀI TẬP 9 - MẢNG

Xóa tất cả các phần tử trùng với x. x do người dùng nhập vào từ bàn phím

1. Đếm số lượng phần tử trùng với x để tính lại kích thước cho mảng tạm
2. Duyệt mảng chính và phần tử nào $\neq x$ thì cho vào mảng tạm



BÀI TẬP 10

Viết chương trình thực hiện những yêu cầu sau:

- Tạo ngẫu nhiên mảng một chiều n phần tử nguyên có giá trị chứa trong đoạn $[-100, 100]$ và xuất mảng. Đếm số phần tử chia hết cho 4 và có chữ số tận cùng là 6.
- Thay các phần tử lẻ bằng 2 lần giá trị của nó.

```
Nhập n = 10
70 -67 22 -87 34 16 -34 -58 76 -78
Có 2 phần tử chia hết cho 4, tận cùng 6
Nhân đôi phần tử lẻ:
70 -134 22 -174 34 16 -34 -58 76 -78
```

BÀI TẬP 12

Tạo ngẫu nhiên mảng một chiều n phần tử nguyên có giá trị chứa trong đoạn $[-100, 100]$ và xuất mảng. Xuất ra màn hình “run” tăng dài nhất tìm thấy đầu tiên.



“run” là chuỗi các phân tử liên tục theo cùng một quy luật nào đó (tăng dần, giảm dần, chẵn, lẻ, bằng nhau,...)

Nhập n : 10

-53 -32 23 78 61 -1 95 83 -55 -7

“run” tăng dài nhất: -53 -32 23 78

BÀI TẬP 13

Tạo mảng một chiều n phần tử nguyên có giá trị nhập vào từ bàn phím. Thống kê số lần xuất hiện của các phần tử trong mảng.

Ví dụ: 1 2 2 3 4 3 2 5 5 3

1[1] 2 [3] 3[3] 4[1] 5[2]

1 3 0 0 0 0 0 0 0 0

CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

BÀI TẬP 14

Viết chương trình thực hiện những yêu cầu sau:

- Tạo mảng một chiều n phần tử nguyên có giá trị nhập vào từ bàn phím. Kiểm tra xem mảng có đối xứng hay không.
- Hãy dịch trái xoay vòng mảng k lần, k nhập từ bàn phím.

Ví dụ: 1 2 3 4 5 5 4 3 2 1

Đối xứng

Nhập số lần cần dịch: 3

4 5 5 4 3 2 1 1 2 3

-3	6	33	12	-8	11
----	---	----	----	----	----

-3	6	33	12	-8	11
----	---	----	----	----	----