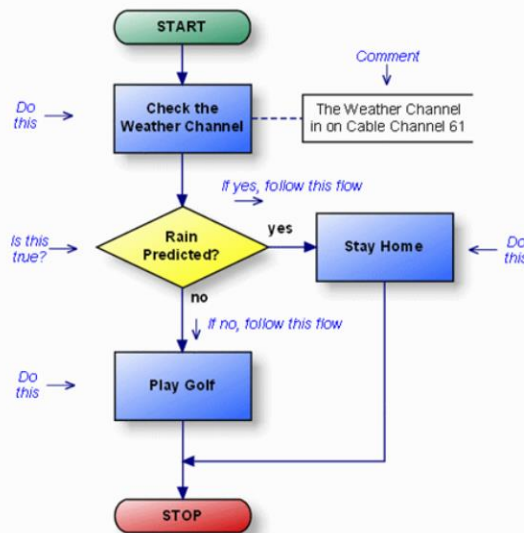


# NHẬP MÔN LẬP TRÌNH

## CHƯƠNG 5: MẢNG MỘT CHIỀU (ARRAY)



ThS: Phạm Nguyễn Sơn Tùng

Email: [pnstung@fit.hcmus.edu.vn](mailto:pnstung@fit.hcmus.edu.vn)

# THÔNG TIN CHUNG

**1**

**Giới thiệu tổng quan**

**2**

**Các cách khai báo mảng một chiều**

**3**

**Các kỹ thuật xử lý trên mảng một chiều**

**4**

**Bài tập ứng dụng tại lớp**

**5**

**Bài tập về nhà**

# GIỚI THIỆU TỔNG QUAN MẢNG MỘT CHIỀU

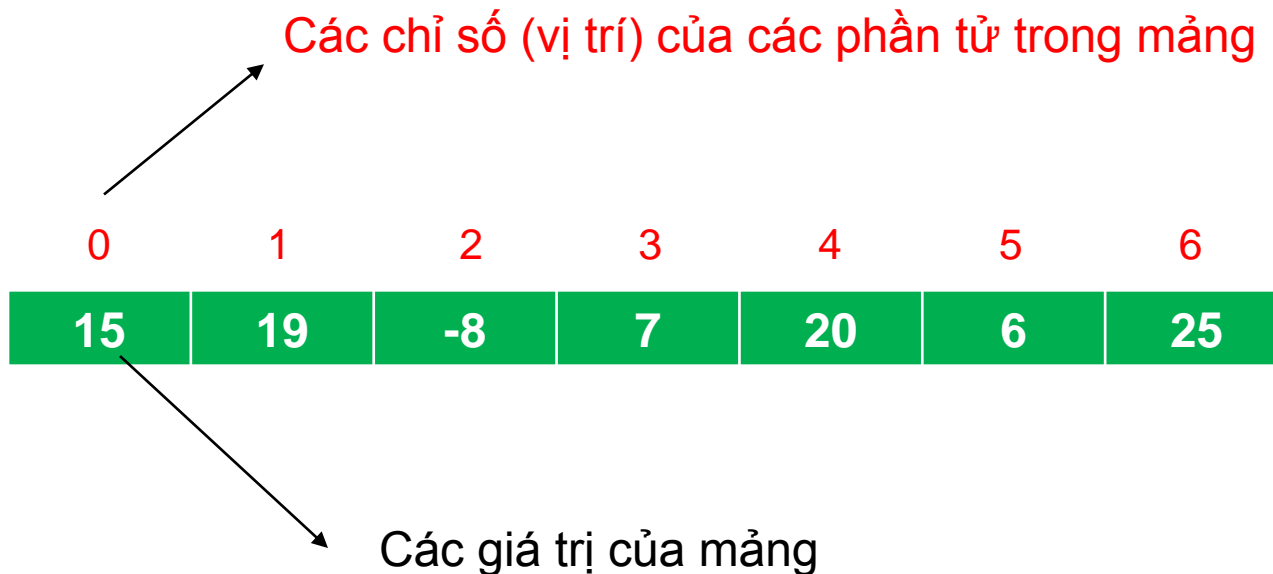
**Định nghĩa:** Mảng một chiều là một cấu trúc dữ liệu, dùng để biểu diễn các đối tượng ở dạng một dãy các thành phần, cùng kiểu dữ liệu với nhau.

Ví dụ:

- Dãy các số nguyên: 2, 8, -3, 4
- Dãy các số thực: 2.4, -1.8, 4.5

# HÌNH ẢNH MINH HỌA MẢNG MỘT CHIỀU

- Cho mảng một chiều **số nguyên** có **7** phần tử:



# CÁC CÁCH KHAI BÁO MẢNG MỘT CHIỀU

- Có 2 cách khai báo để sử dụng mảng một chiều:

Sử dụng mảng tĩnh

Sử dụng mảng động

# CÁC CÁCH KHAI BÁO MẢNG MỘT CHIỀU

- Có 2 cách khai báo để sử dụng mảng một chiều:

## Sử dụng mảng tĩnh

- Mảng tĩnh là mảng cấp phát bộ nhớ ngay từ đầu.

## Sử dụng mảng động

- Mảng động là mảng sẽ được cấp phát vùng nhớ trong khi chương trình chạy.

# KHAI BÁO MẢNG TĨNH

0	1	2	3	4	5	6
15	19	-8	7	20	6	25

Kiểu dữ liệu  
Tên của mảng  
Số lượng phần tử có trong mảng

```
int a[20];  
a[0] = 15;  
a[1] = 19;  
a[2] = -8;  
...
```

# TRUY XUẤT VÀ THAO TÁC

0	1	2	3	4	5	6
15	19	-8	7	20	6	25

```
//Truy xuất phần tử trong mảng
```

```
int x = a[1];
```

```
int y = a[6];
```

```
//Thao tác phần tử trong mảng
```

```
a[2] = 5;
```

```
a[3] = 9;
```

```
//Sai - báo lỗi
```

```
a[-1] = 7
```



# KHAI BÁO MẢNG TĨNH

```
#include <iostream>
using namespace std;
#define MAX 100

void NhapMang(int a[], int& n)
{
    cout<<"Nhap so luong phan tu n: ";
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        cout<<"Nhap phan tu thu "<<i <<": ";
        cin>>a[i];
    }
}
```

# KHAI BÁO MẢNG TĨNH

```
void XuatMang(int a[], int n)
{
    cout<<"Nội dung của mảng là: ";
    for (int i = 0; i < n; i++)
        cout << a[i] << " ";
}

int main()
{
    int n;
    int a[MAX];
    NhapMang(a, n);
    XuatMang(a, n);
    return 0;
}
```

# CÁC KỸ THUẬT XỬ LÝ MẢNG

- 1 Kỹ thuật đặt lính canh
- 2 Kỹ thuật liệt kê
- 3 Kỹ thuật tính tổng
- 4 Kỹ thuật đếm số phần tử
- 5 Kỹ thuật đặt cờ hiệu
- 6 Kỹ thuật sắp xếp mảng
- 7 Kỹ thuật thêm phần tử vào mảng
- 8 Kỹ thuật xóa phần tử trong mảng
- 9 Kỹ thuật xây dựng mảng
- 10 Kỹ thuật xử lý mảng và mảng con

# 1. KỸ THUẬT ĐẶT LÍNH CANH

**Định nghĩa:** là kỹ thuật dùng để tìm kiếm một giá trị (phần tử) nào đó thỏa mãn điều kiện hay ràng buộc theo quy định.



# KỸ THUẬT ĐẶT LÍNH CANH

**Ví dụ thực tế:** Hãy tìm sinh viên có chiều cao, cao nhất trong lớp học này.

- Chọn bất kỳ một sinh viên nào đó làm lính canh và cho em sinh viên này là sinh viên cao nhất trong lớp.
- Sinh viên này đứng trước cửa lớp, từng sinh viên khác đi ra khỏi lớp và so sánh với em này. Em nào nhỏ hơn thì cho qua, nếu lớn hơn thì thay thế em đang đứng làm lính canh.
- Tiếp tục cho đến khi tất cả sinh viên ra khỏi lớp học.

# KỸ THUẬT ĐẶT LÍNH CÁN

- ❑ **Bài toán minh họa:** Cho mảng một chiều **số nguyên** viết hàm tìm giá trị lớn nhất:

0	1	2	3	4	5	6
15	19	-8	7	20	6	2

- Max = 15
- So sánh  $a[1]$  và Max  $\rightarrow$  Max = 19.
- So sánh  $a[2]$  và Max
- So sánh  $a[3]$  và Max
- So sánh  $a[4]$  và Max  $\rightarrow$  Max = 20.
- So sánh  $a[5]$  và Max
- So sánh  $a[6]$  và Max

# CODE MINH HỌA

```
int TimLonNhat(int a[], int n)
{

}
}
```

# BÀI TẬP KỸ THUẬT ĐẶT LÍNH CANH

**Bài tập 1:** Tìm số chẵn cuối cùng trong mảng một chiều các số nguyên. Nếu mảng không có giá trị chẵn thì trả về giá trị -1.

**Bài tập 2:** Tìm giá trị dương nhỏ nhất trong mảng một chiều các số nguyên. Nếu mảng không có giá trị dương thì trả về giá trị -1.

**Bài tập 3:** Tìm số nguyên tố đầu tiên trong mảng một chiều các số nguyên. Nếu mảng không có số nguyên tố thì trả về giá trị -1.



## 2. KỸ THUẬT LIỆT KÊ

**Định nghĩa:** là kỹ thuật dùng để liệt kê ra các giá trị (phần tử) thỏa mãn tính chất nào đó.



# KỸ THUẬT LIỆT KÊ

**Ví dụ thực tế:** Hãy liệt kê những con cá không bị ươn trong rổ cá này.

- Tìm hiểu làm thế nào để phân biệt được những con cá không bị ươn là những con cá có biểu hiện gì.
- Xem qua từng con cá nếu con cá nào thỏa mãn tính chất trên thì chọn ra riêng một bên.
- Tiếp tục tìm và chọn cho đến khi hết rổ cá.

# KỸ THUẬT LIỆT KÊ

**Bài toán minh họa:** Cho mảng một chiều **số nguyên**. Viết hàm liệt kê các số chẵn trong mảng:

0	1	2	3	4	5	6
15	19	-4	8	20	6	7

Các giá trị chẵn trong mảng là:

■ Kết quả: -4, 8, 20, 6

# CODE MINH HỌA

```
void LietKeChan(int a[], int n)
{

}
}
```

# BÀI TẬP KỸ THUẬT LIỆT KÊ

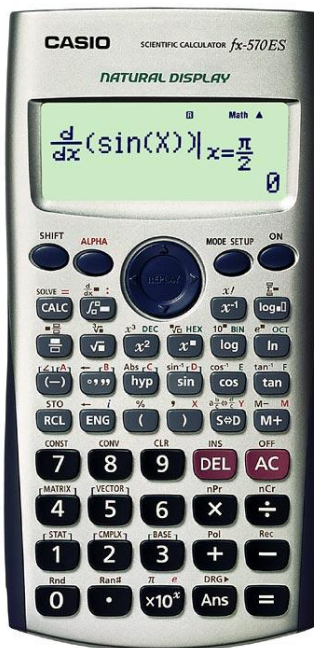
**Bài tập 1:** Hãy liệt kê các vị trí mà tại đó chứa giá trị lớn nhất trong mảng một chiều các số thực.

**Bài tập 2:** Hãy liệt kê các giá trị trong mảng một chiều các số nguyên có chữ số đầu tiên là số lẻ.

**Bài tập 3:** Hãy liệt kê các vị trí mà tại đó chứa số chính phương trong mảng một chiều các số nguyên.

### 3. KỸ THUẬT TÍNH TỔNG

**Định nghĩa:** là kỹ thuật dùng để tính tổng các phần tử của mảng theo một quy tắc/quy định nào đó.



# KỸ THUẬT TÍNH TỔNG

**Ví dụ thực tế:** Hãy tính tổng tiền các sản phẩm là rau trong giỏ hàng mua tại siêu thị.

- Lựa ra trong giỏ hàng những sản phẩm nào là rau.
- Lần lượt cộng tất cả giá tiền của các sản phẩm đã lựa xong lại ta được tổng tiền cuối cùng là kết quả cần lấy.
- Cần thận đừng để số tiền chi trả vượt quá số tiền trong túi.

# KỸ THUẬT TÍNH TỔNG

**Bài toán minh họa:** Cho mảng một chiều **số nguyên**.  
Viết hàm tính tổng các số nguyên tố trong mảng:

0	1	2	3	4	5	6
1	5	2	13	8	9	7

▣ Tổng các số nguyên tố trong mảng là:

$$\rightarrow \text{Tong} = 5 + 2 + 13 + 7 = 27$$



# CODE MINH HỌA

```
int TongNguyenTo(int a[], int n)
{

}
}
```

# BÀI TẬP KỸ THUẬT TÍNH TỔNG

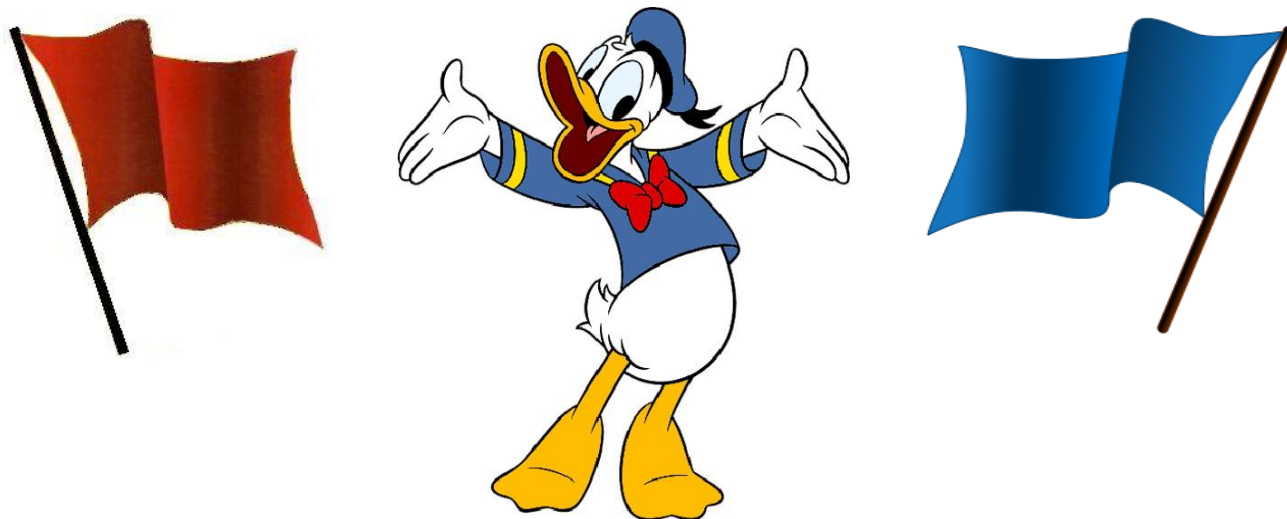
**Bài tập 1:** Tính tổng các giá trị có chữ số hàng chục là chữ số 7 trong mảng một chiều các số nguyên.

**Bài tập 2:** Tính tổng các phần tử có giá trị lớn hơn giá trị đứng liền trước nó trong mảng một chiều các số thực.

**Bài tập 3:** Tính tổng các giá trị trong mảng một chiều các số nguyên. Mà giá trị đó chứa toàn số chẵn.

## 4. KỸ THUẬT ĐẶT CỜ HIỆU

**Định nghĩa:** là kỹ thuật dùng để kiểm tra xem mảng có thỏa mãn tính chất nào đó hay không. Nếu có cờ **True** sẽ bật lên, ngược lại nếu không thỏa mãn cờ **False** sẽ bật lên.



# KỸ THUẬT ĐẶT CỜ HIỆU

- ❑ **Ví dụ thực tế:** Kiểm tra xem có phải tất cả chiếc xe máy trong cửa hàng đều xuất xứ từ Nhật.
  - Kiểm tra lần lượt từng chiếc xe máy.
  - Chỉ cần một chiếc xe máy không xuất phát từ Nhật thì xem như không thỏa mãn điều kiện đưa ra.

# KỸ THUẬT ĐẶT CỜ HIỆU

**Bài toán minh họa:** Cho mảng một chiều **số thực**. Kiểm tra xem mảng có chứa toàn số dương hay không:

0	1	2	3	4	5	6
15.7	19	-4.1	8	10	-6	8.4

Đặt cờ hiệu ban đầu:

- Flag = **True**
- Kiểm tra lần lượt tất cả các phần tử trong mảng, thấy phần tử -4.1 nằm ở vị trí thứ 2 không thỏa mãn.
- Flag = **False**

# CODE MINH HỌA

```
bool KiemTraToanDuong(float a[], int n)
{

}
}
```

# BÀI TẬP KỸ THUẬT ĐẶT CỜ HIỆU

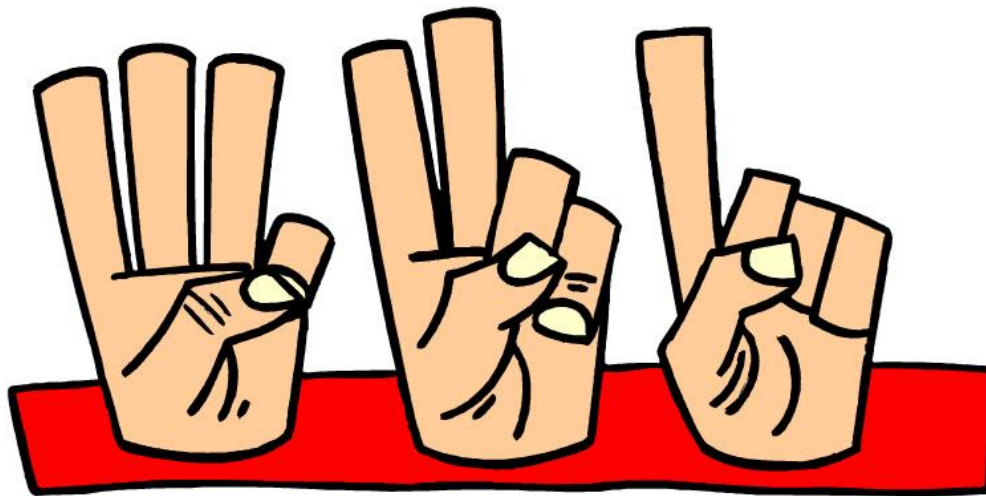
**Bài tập 1:** Kiểm tra xem mảng **số nguyên** có tính **chất chẵn lẻ** hay không. Tính chất chẵn lẻ là: có một số chẵn và một số lẻ xen kẽ với nhau (Có trả về 1, không trả về 0)

**Bài tập 2:** Kiểm tra xem mảng **số thực** có tồn tại **phần tử cực tiểu** hay không? Phần tử cực tiểu là phần tử nhỏ hơn các phần tử xung quanh nó. (có trả về 1, không trả về 0).

**Bài tập 3:** Hãy kiểm tra xem mảng **số nguyên** có tồn tại **2 giá trị giống nhau liên tiếp** hay không?

## 5. KỸ THUẬT ĐẾM SỐ PHẦN TỬ

**Định nghĩa:** là kỹ thuật dùng để **đếm** số phần tử thỏa mãn tính chất nào đó.





# KỸ THUẬT ĐẾM

**Ví dụ thực tế:** Hãy đếm xem có bao nhiêu chai nước suối Aquafina trong tủ lạnh.

- Đếm ban đầu là 0.
- Kiểm tra từng chai nước suối một, mỗi lần tìm một chai nước suối Aquafina thì tang lên 1.
- Lần lượt đếm tất cả cho đến khi hết nước suối trong tủ lạnh, ta được con số cuối cùng là kết quả cần lấy.

# KỸ THUẬT ĐẾM

**Bài toán minh họa:** Cho mảng một chiều **số nguyên**.  
Viết hàm đếm các phần tử chia hết cho 7:

0	1	2	3	4	5	6
15	19	-4	14	17	-6	7

▣ Các giá trị chi hết cho 7 trong mảng: 14, 7

■ Đếm = 2

# CODE MINH HỌA

```
int DemChiaHet7(int a[], int n)
{

}
}
```

# BÀI TẬP KỸ THUẬT ĐẾM

**Bài tập 1:** Đếm tần suất xuất hiện của số  $x$  trong mảng một chiều **số nguyên**,  $x$  do người dùng nhập vào.

**Bài tập 2:** Đếm số lượng các giá trị xuất hiện một lần trong mảng một chiều các **số thực**.

**Bài tập 3:** Đếm số lượng các phần tử có giá trị khác nhau trong mảng một chiều các **số thực**.

■ Ví dụ: Đếm = 3

7	3	32	7
---	---	----	---

## 6. KỸ THUẬT SẮP XẾP MẢNG

**Định nghĩa:** là kỹ thuật dùng để sắp xếp mảng có thứ tự **giảm dần** hoặc **tăng dần** có thể là cả mảng hoặc một vài phần tử trong mảng.



# KỸ THUẬT SẮP XẾP

- ❑ **Ví dụ thực tế:** Hãy sắp xếp các sinh viên trong lớp theo điểm trung bình tăng dần.
  - Nhìn tổng điểm trung bình của cả lớp.
  - Xem điểm sinh viên nào thấp nhất xếp đầu tiên, lần lượt nhìn các điểm trung bình của các sinh viên còn lại (trừ sinh viên đã chọn ban đầu) sinh viên nào thấp nhất để vào vị trí kế bên sinh viên ban đầu.
  - Lần lượt xét cho đến hết các sinh viên trong lớp ta sẽ sắp xếp được theo ý muốn.

# KỸ THUẬT SẮP XẾP MẢNG

**Bài toán minh họa:** Sắp xếp mảng các số nguyên sau theo thứ tự **tăng dần**:

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

<https://www.youtube.com/watch?v=ROaIU379I3U>

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

# CODE MINH HỌA

```
void insertionSort(int a[], int n)
{

}
}
```



# BÀI TẬP KỸ THUẬT SẮP XẾP

**Bài tập 1:** Hãy sắp xếp các số **nguyên tố** tăng dần các số khác vẫn giữ nguyên vị trí trong mảng một chiều các **số nguyên**.


**Bài tập 2:** Hãy sắp xếp các số **âm giảm** dần và số dương **tăng dần** trong mảng một chiều các **số thực**.

## 7. KỸ THUẬT THÊM PHẦN TỬ VÀO MẢNG


**Định nghĩa:** là kỹ thuật dùng để thêm bất kỳ một phần tử nào đó ở ngoài vào bên trong mảng, vị trí thêm vào cũng bất kỳ nơi nào đó trong phạm vi của mảng.

92	89	96	95	
Score[0]	Score[1]	Score[2]	Score[3]	Score[4]

92	89	96		95
Score[0]	Score[1]	Score[2]	Score[3]	Score[4]

92	89		96	95
Score[0]	Score[1]	Score[2]	Score[3]	Score[4]

92	89	93	96	95
Score[0]	Score[1]	Score[2]	Score[3]	Score[4]

# KỸ THUẬT THÊM PHẦN TỬ VÀO MẢNG

- **Ví dụ thực tế:** Hãy thêm một bạn sinh viên “Nguyễn Văn Tèo” ngồi vào bàn trong lớp.
  - Xét bàn mà sinh viên này sẽ ngồi vào.
    - Dọn một chỗ trống cho sinh viên này.
    - Xếp sinh viên vào chỗ vừa dọn xong ở bước trên.

# KỸ THUẬT THÊM PHẦN TỬ VÀO MẢNG

- ❑ **Bài toán minh họa:** Cho mảng một chiều **số nguyên**. Hãy thêm phần tử có giá trị là 3 vào vị trí 4 của mảng.

0	1	2	3	4	5	6
15	19	-4	14	17	-6	7

- ❑ Tăng kích thước mảng lên 1 đơn vị:
- ❑ Dời 3 số 17, -6, 7 về phía sau 1 đơn vị:
- ❑ Thêm số 3 vào vị trí 4.

# CODE MINH HỌA

```
void ThemPhanTu(int a[], int& n, int val, int pos)
{

}
}
```

# BÀI TẬP KT THÊM PHẦN TỬ

**Bài tập 1:** Hãy thêm phần tử  $x$  vào phía sau các phần tử **lớn nhất** trong mảng một chiều các **số nguyên**.

## 8. KỸ THUẬT XÓA PHẦN TỬ TRONG MẢNG

**Định nghĩa:** là kỹ thuật dùng để xóa bất kỳ một phần tử nào đó của mảng, vị trí của phần tử bị xóa là bất kỳ nơi nào đó trong phạm vi của mảng.



# KỸ THUẬT XÓA PHẦN TỬ TRONG MẢNG

- **Ví dụ thực tế:** Hãy dời (xóa) sinh viên “Nguyễn Văn Tí” đang ngồi trong bàn học.
  - Xét bàn mà sinh viên Tí đang ngồi.
    - Mời sinh viên ra khỏi bàn học.
    - Sắp xếp lại các vị trí ngồi của bàn.



# KỸ THUẬT XÓA PHẦN TỬ TRONG MẢNG

**Bài toán minh họa:** Cho mảng một chiều **số nguyên**. Hãy xóa phần tử ở vị trí 2.

0	1	2	3	4	5	6
15	19	-4	14	17	-6	7

- ❑ Dời các phần tử từ vị trí cần xóa lên trên
  - ❑  $a[i] = a[i+1]$
- ❑ Giảm kích thước của mảng đi một:
  - ❑  $n = n - 1$

# CODE MINH HỌA

```
void XoaPhanTu(int a[], int& n, int pos)
{

}
}
```

# BÀI TẬP KT XÓA PHẦN TỬ

**Bài tập 1:** Xóa tất cả các số âm trong mảng một chiều các số nguyên.

**Bài tập 2:** Xóa tất cả các phần tử có tần xuất xuất hiện nhiều hơn một lần trong mảng các số thực.

**Bài tập 3:** Xóa các phần tử dư thừa để mảng trở thành mảng toàn các số dương. Mảng số nguyên.

## 9. KỸ THUẬT XÂY DỰNG MẢNG

**Định nghĩa:** là kỹ thuật dùng tạo ra một mảng mới từ mảng ban đầu đã cho, với các phần tử trong mảng mới được sao chép từ mảng ban đầu theo một quy định nào đó.



# KỸ THUẬT XÂY DỰNG MẢNG

**Bài toán minh họa:** Cho mảng một chiều a **số nguyên**. Hãy xây dựng mảng b là các phần tử a chia hết cho 2.

	0	1	2	3	4	5	6
Mảng a	15	19	-4	14	17	-6	7

	0	1	2
Mảng b	-4	14	-6

# CODE MINH HỌA

```
void XayDungMang(int a[], int n, int b[], int& m)
{

}
}
```

# BÀI TẬP KT XÂY DỰNG MẢNG

**Bài tập 1:** Cho mảng một chiều các số nguyên  $a$ , hãy tạo ra mảng  $b$  từ mảng  $a$ , với  $b[i]$  là các phần tử đầu cuối mảng  $a$ , kéo dần đến phần tử giữa mảng.

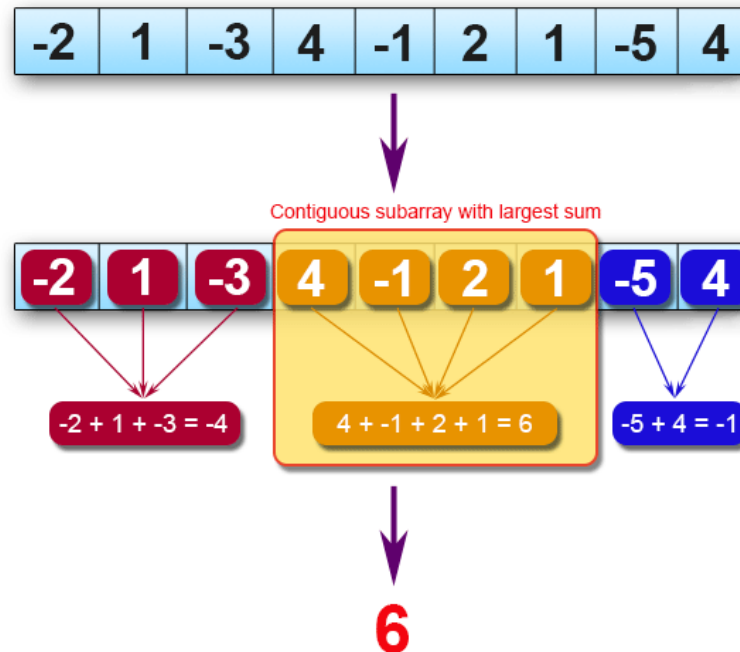
**Bài tập 2:** Hãy tạo ra mảng  $a$  chứa các số nguyên, biết các phần tử của  $a$  là ngẫu nhiên (random).

**Bài tập 3:** Hãy tạo ra mảng  $a$  các số nguyên ngẫu nhiên trong đoạn  $[min, max]$  cho trước.

**Bài tập 4:** Cho mảng một chiều các số nguyên  $a$ , hãy tạo ra mảng  $b$  từ mảng  $a$ , với  $b[i]$  là các phần tử là số đầu tiên của mảng  $a$ .

# KỸ THUẬT XỬ LÝ MẢNG VÀ MẢNG CON

**Định nghĩa:** Trong mảng cho ban đầu có nhiều mảng con nhỏ, mỗi mảng con có tính chất khác nhau, kỹ thuật này dùng để xử lý các mảng con đó.





# KỸ THUẬT XỬ LÝ MẢNG CON

**Bài toán minh họa:** Cho mảng một chiều a **số nguyên**. Hãy liệt kê các mảng con có 2 phần tử chẵn liên tiếp nhau.

0	1	2	3	4	5	6
15	19	-4	14	8	-6	7

Kết quả:

2 3

3 4

4 5

# CODE MINH HỌA

```
void LietKeMangCon(int a[], int n)
{

}
}
```

# KỸ THUẬT XỬ LÝ MẢNG VÀ MẢNG CON

**Bài tập 1:** Hãy đảo ngược  $\frac{1}{2}$  mảng ban đầu, mảng các số nguyên.

**Bài tập 2:** Liệt kê tất cả các mảng con có độ dài lớn hơn 2 trong mảng một chiều các số nguyên.