

# Bài tập buổi 4

## ***Yêu cầu trước khi làm:***

- Nộp file.c
- Định dạng: **STT\_HoVaTen\_Bai1.c**
- Ví dụ: **S1\_01\_NguyenVietTrung\_Bai1.c**
- Bạn nào làm bằng điện thoại thì chụp ảnh màn hình mà chèn thêm tên vào góc phải bên dưới ảnh.
- Vận dụng các kiến thức đã học:
  - câu lệnh nhập xuất::printf, scanf
  - câu lệnh rẽ nhánh: if, else, if else,...
  - Vòng lặp: for, while, do while
  - Giới thiệu break, continue
  - Mảng 1 chiều và Các bài toán trên mảng
- Nếu có gì không hiểu có thể hỏi anh chị support nhé.

## ***Tìm hiểu các kiến thức cho buổi sau:***

- Function: tham chiếu(toán tử \*), tham trị (**Phần này khó, yêu cầu tìm hiểu kỹ**)
- Tổng quan về C++
  - Giới thiệu thư viện iostream (cin, cout, endl)
  - Giới thiệu keyword "namespace"
- Kiểu dữ liệu: bool
- Cấu trúc điều khiển rẽ nhánh
  - If, else, toán tử 3 ngôi

## ***Các quy tắc đặt tên:***

<https://www.facebook.com/groups/c16.hit/permalink/364258842128164/>

## Lý thuyết về mảng buổi 4:

```
/*
    Kieu_Du_lieu ten_Mang[So_Luong_Phan_Tu_Toai_Da];
    Kieu_Du_lieu ten_Mang[So_Luong_Phan_Tu_Toai_Da] = {pt1, pt2, ..., ptn};
    Truy xuất phần tử trong mảng: ten_Mang[index]. Ví dụ: a[1];
    index bắt đầu từ 0
    viTri = chỉSo + 1
    mảng là dãy các phần tử liên tiếp nhau

    //Các bài toán liên quan đến mảng: duyệt mảng, sửa, thêm, xóa, sắp xếp...
*/
```

### Khởi tạo mảng và các biến liên quan:

```
int a[100];
int i, n;
```

### Nhập số lượng phần tử mảng và nhập mảng:

```
printf("Nhập số lượng phần tử: ");
scanf("%d", &n); // nhập số lượng phần tử

for (i = 0; i < n; i++) {
    printf("Nhập a[%d] = ", i); // thông báo người dùng nhập phần tử nào
    scanf("%d", &a[i]);         // nhập từng phần tử trong mảng
}
```

### Xuất mảng:

```
printf("Mảng vừa nhập là: ");
for (i = 0; i < n; i++) {
    printf("%d ", a[i]); // xuất từng phần tử trong mảng
}
```

### Bài tập tính tổng hoặc tích các phần tử trong mảng:

```
// Tính tổng và tích các phần tử trong mảng:
int tong = 0, tich = 1;
for (i = 0; i < n; i++) {
    tong += a[i]; // tong = tong + a[i];
    tich *= a[i]; // tich = tich * a[i];
}

printf("\nTong = %d", tong);
printf("\nTich = %d", tich);
```

## Bài tập in ra tất cả các số chẵn trong mảng

```
// in tat ca so chan trong mang
printf("\nCac so chan trong mang: ");
for (i = 0; i < n; i++) {
    if (a[i] % 2 == 0) {
        // so chan
        printf("%d ", a[i]);
    }
}
```

## Bài tập sửa tất cả các phần tử lẻ trong mảng thành giá trị 10

```
// Sửa tất cả các phần tử lẻ trong mảng thành số 10
for (i = 0; i < n; i++) {
    if (a[i] % 2 != 0) {
        a[i] = 10;
    }
}
```

## Bài tập thêm một phần tử vào mảng một chiều

```
/* Với:
    MAX là So_Luong_Phan_Tu_Toai_Da của mảng
    pos là vị trí muốn thêm
    val là giá trị muốn thêm
*/
// Mang da day, khong the them.
if (n >= MAX) {
    printf("Mang da day, khong the them.\n");
} else {
    // Neu pos <= 0 => Them vao dau
    if (pos < 0) {
        pos = 0;
    }
    // Neu pos >= n => Them vao cuoi
    else if (pos > n) {
        pos = n;
    }
    // Dich chuyen mang de tao o trong truoac khi them.
    for (i = n; i > pos; i--) {
        a[i] = a[i - 1];
    }
    // Chen val tai pos
    a[pos] = val;
    // Tang so luong phan tu sau khi chen.
    ++n;
}
```

## Bài tập xóa một phần tử trong mảng một chiều:

```
// Mảng rỗng, không thể xóa.
if (n <= 0) {
    printf("Mảng rỗng, không thể xóa.\n");
} else {
    // Nếu pos <= 0 => Xóa đầu
    if (pos < 0) {
        pos = 0;
    }
    // Nếu pos >= n => Xóa cuối
    else if (pos >= n) {
        pos = n - 1;
    }
    // Dịch chuyển mảng
    for (int i = pos; i < n - 1; i++) {
        a[i] = a[i + 1];
    }
    // Giảm số lượng phần tử sau khi xóa.
    --n;
}
```

## Bài tập sắp xếp mảng:

<https://www.facebook.com/100051093398680/videos/471944677699275/>

```
// sắp xếp giảm dần:
for (i = 0; i < n; i++) {
    for (j = i + 1; j < n; j++) {
        if (a[i] < a[j]) {
            int temp = a[i];
            a[i] = a[j];
            a[j] = temp;
        }
    }
}
```

## Bài tập về nhà buổi 4:

**Bài 1:**(25đ) Nhập vào một mảng a gồm **n số nguyên** ( $n > 0 \ \&\& \ n < 100$ ). Kiểm tra xem mảng đang là mảng giảm dần hay tăng dần. In ra “**YES**” nếu mảng là **mảng tăng dần**, in ra “**NO**” nếu mảng là **mảng giảm dần**. Nếu **không tăng không giảm** thì in ra -1

**Input:**

- Dòng thứ nhất là số lượng phần tử trong mảng
- Dòng thứ 2 là các phần tử trong mảng

**Output:** In ra đáp án theo yêu cầu của đề bài

Note: tăng dần và giảm dần thì đều hiểu là tăng hay giảm 1 hoặc nhiều đơn vị

Input	Output
4 1 2 3 4	YES
5 7 6 5 4 3	NO
4 1 7 4 9	-1

**Bài 2:** (20đ) Nhập vào một mảng gồm **n số thực**( $n > 0 \ \&\& \ n < 100$ ). Sắp xếp các số **âm** theo chiều giảm dần, các số khác thì đứng yên. In mảng đã sắp xếp ra màn hình.

Input	Output
6 -4 3 6.7 -1 0 1.3	-1 3 6.7 -4 0 1.3

- có 2 số âm là -4 và -1, thay đổi vị trí 2 số để thỏa mãn âm theo chiều giảm dần, các số khác không thay đổi.

**Bài 3:** (20đ) Nhập vào một mảng gồm **n số nguyên** ( $n > 0 \ \&\& \ n < 100$ ).

- Đếm và in ra các **số chính phương** trong mảng. Các số chính phương là **các số căn bậc 2 của nó là số nguyên**.
- Đếm và in ra các **số hoàn hảo** trong mảng. **Số hoàn hảo là số mà tổng các ước bằng 2 lần chính nó**.

Input	Output
5 2 3 4 5 6	1 4
5 1 2 6 28 4	2 6 28

- test 1: căn bậc 2 của 4 là 2 (số nguyên) -> 4 là số chính phương
- test 2: 6 có ước là 1 2 3 6, tổng các ước là  $12 = 2 \cdot 6$  -> 6 là số hoàn hảo. Tương tự với số 28.

**Bài 4:** (15đ) Nhập vào một mảng gồm n số **thực** ( $n > 0 \ \&\& \ n < 100$ ). In ra các **vị trí** mà **giá trị tại đó là lớn nhất** trong mảng.

Input	Output
6 1.4 5 4 2.6 3 5	2 6

**Bài 5:** (20đ) Nhập vào một mảng gồm n số **dương** n ( $n > 0 \ \&\& \ n < 100$ ). Tìm các **số nguyên tố** có trong mảng, tính tổng các số nguyên tố đó. Biết số nguyên tố là số có **2 ước** là 1 và chính nó.

Input	Output
5 2 3 4 5 6	2 3 5 10

- 2 có ước là 1 và 2 -> 2 là số nguyên tố. Tương tự 3 và 5.