**Bài tập buổi 5**

***Yêu cầu trước khi làm:***

* Nộp file.c và file.cpp
* Đối với file c: **STT\_HoVaTen\_Bai1.c**

Ví dụ: **S1\_01\_NguyenVietTrung\_Bai1.c**

* Đối với file cpp: **STT\_HoVaTen\_Bai2.cpp**

Ví dụ: **S1\_01\_NguyenVietTrung\_Bai2.cpp**

* 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
* Function: tham chiếu(toán tử \*), tham trị
* Tổng quan về C++:
  + Giới thiệu thư viện iostream (cin, cout, endl)
  + Giới thiệu keyword “namespace”
  + Kiểu dữ liệu khác: bool, string
* Nếu có gì không hiểu có thể hỏi anh chị support nhé.

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

* Vòng lặp, break, continue
* Mảng
* Các bài toán trên mảng
* Function: Tham chiếu, tham trị

***Các quy tắc đặt tên***: <https://www.facebook.com/groups/c16.hit/permalink/364258842128164/>

***Lý thuyết về mảng buổi 5:***

**Hàm:**

**// Hàm (Function)**

**/\* Các loại:**

**1. Hàm thư viện/hàm chuẩn:**

**stdio.h: printf(), scanf(), gets(), puts(),…**

**math.h : abs(), sqrt(), pow(), sin(), cos(),…**

**...**

**2. Hàm do người dùng định nghĩa**

**\*Cú pháp khai báo hàm:**

**<kiểu dữ liệu> <tên hàm> (<các tham số>){**

**// thân hàm**

**return <value>; // value phải cùng kiểu <kiểu dữ liệu>**

**}**

**\* Ví dụ:**

**int Add(int *a*, int *b*){**

**int sum = a + b;**

**return sum;**

**}**

**\*/**

**Các kiểu hàm do người dùng tự định nghĩa:**

Xem chi tiết tại: [types-user-defined-functions](https://www.programiz.com/c-programming/types-user-defined-functions)

* Hàm trả về giá trị: void
* Hàm không trả về giá trị: int, long, float, double, …

**Tham chiếu và tham trị: Tham khảo thêm: tại** [**viettuts.vn**](https://viettuts.vn/lap-trinh-c/truyen-tham-chieu-va-tham-tri-trong-c)

**- Con tro: int \*a; them dau \* truoc ten bien**

**- Tham tri:**

**+) truyen gia tri vao trong ham**

**+) khong lam thay doi gia tri cua bien truyen vao ham day**

**- Tham chieu:**

**+) truyen dia chi o nho: toan tu &**

**+) Lam thay doi gia tri cua bien truyen vao**

**+) Doi so cua ham la con tro: \***

**+) Truy xuat hay thay doi gia tri con tro thi dung toan tu \***

**Ví dụ hàm nhập mảng:**

**void NhapMang(int *a*[], int *n*) {**

**int i;**

**printf("Nhap vao mang %d so nguyen: ", n);**

**for (i = 0; i < n; i++)**

**scanf("%d", &a[i]);**

**}**

**Ví dụ hàm xuất mảng:**

**void XuatMang(int *a*[], int *n*) {**

**int i;**

**printf("Mang vua nhap la: ");**

**for (i = 0; i < n; i++)**

**printf("%d ", a[i]);**

**}**

**Ví dụ tính tổng 2 số nguyên:**

**int Sum(int *a*, int *b*) {**

**// int s = a + b;**

**// return s;**

**return a + b;**

**}**

**Ví dụ hàm không trả về giá trị và không có đối số**

**void ShowInfo() {**

**printf("\nNguyen Dinh Huan");**

**}**

**Ví dụ hàm tham chiếu**

**void Show(int \**a*) {**

**printf("\na trong ham truoc khi thay doi gia tri = %d", \*a);**

**\*a = 5;**

**printf("\na trong ham sau khi thay doi gia tri = %d", \*a);**

**}**

***Bài tập về nhà buổi 5:***

**Yêu cầu tất cả đều dùng hàm (hàm nhập /xuất của mảng , dùng hàm xử lý bài toán, gọi các hàm trên trong hàm int main())**

**Truyền tham chiếu biến vào hàm (dùng \*ten\_bien) (nếu cần thiết)**

**Bài 1(25):** Sử dụng hàm nhập, xuất mảng **n** số nguyên **dương** (0<n<100). Sau đó **xóa** các **số đen đủi** có trong mảng. Rồi sắp xếp theo **chiều tăng dần**

**Số đen đủi** là số **chỉ** chứa chữ số 4 hoặc chữ số 7 hoặc chứa cả hai chữ số 4 và 7 trong số đó( ví dụ 4,7,44,47, 447, 777, 77777.......)

Input:

* Dòng đầu là số lượng phần tử trong mảng
* Dòng thứ hai là các phần tử trong mảng

Output:

* Một dòng duy nhất là mảng sau khi xóa các số đen đủi và đã được sắp xếp tăng dần

| Input | Output |
| --- | --- |
| 5  37 2 44 447 6 | 2 6 37 |
| 8  4 44 5 4474 2 43 6 24 | 2 5 6 24 43 |

**Giải thích**: Test case 2: Mảng chứa các số đen đủi là 4 44 4474 thì xóa hết các số này. rồi sắp xếp lại mảng vừa xóa

**Bài 2(50):** Sử dụng hàm nhập, xuất mảng **n** số nguyên (0<n<100)

**a)(25)** Nhập số nguyên **x**, chèn **x** vào **sau vị trí** của **số âm đầu tiên** trong mảng, nếu **không có số âm** thì chèn vào **cuối mảng**

**Input:**

* Dòng đầu là số lượng phần tử trong mảng
* Dòng thứ hai là các phần tử trong mảng
* Dòng thứ ba là số nguyên x cần chèn

**Output:** Một dòng duy là nhất là mảng sau khi chèn x

| Input | Output |
| --- | --- |
| 6  1 4 -5 9 -8 7  3 | 1 4 -5 3 9 -8 7 |
| 6  2 5 4 9 6 7  3 | 2 5 4 9 6 7 3 |

**Giải thích:**

**+) Test case 1:** số nguyên x có giá trị là 3 được chèn sau vị trí của -5( giá trị âm đầu tiên của mảng)

**+) Test case 2:** Mảng không có số âm => chèn 3 vào vị trí cuối cùng của mảng

**b)(25)** Sắp xếp mảng vừa chèn thêm sao cho **phần tử lớn nhất** ở **đầu mảng**, **phần tử bé nhất** ở **cuối** **mảng**, **các phần tử còn lại** sắp xếp **tăng dần**. In mảng đã sắp ra màn hình.

Input: (lấy mảng vừa chèn ở trên làm input)

Output:Một dòng duy là nhất là mảng sau khi sắp xếp lại mảng vừa chèn ở trên theo yêu cầu

| Input | Output |
| --- | --- |
| 7  1 4 -5 3 9 -8 7 | 9 -5 1 3 4 7 -8 |

**Giải thích:** phần tử lớn nhất của mảng là 9 , bé nhất của mảng là -8,

đưa 9 lên trên đầu, -8 xuống cuối và các phần tử khác sắp xếp tăng dần

**Bài 3(25):** Sử dụng hàm nhập, xuất mảng **n**  số nguyên **dương** gồm n phần tử (2<n<=50)

Sửa số tất cả **số may mắn** trong mảng thành giá trị **-1**

(**số may mắn** là số có **tổng các chữ số** của số đó là **một số nguyên tố**)

VD: 12 là số may mắn do tổng các chữ số của số 12 là 1+2=3 mà 3 là số nguyên số,

| Input | Output |
| --- | --- |
| 5  9 5 15 29 26 | 9 -1 15 -1 26 |

**Giải thích**

**Input:**

* **Dòng đầu** là số lượng phần tử trong mảng
* **Dòng thứ hai** là các phần tử trong mảng

**Output**:

* **Mảng mới**: 9 -1 15 -1 26

**Số** 5 , 29 là những **số may mắn** do tổng các chữ số là một số nguyên tố nên ta sửa thành -1