MỘT SỐ CHÚ Ý / MẸO KHI THI NHẬP MÔN LẬP TRÌNH

A. Chuỗi C (mảng ký tự char[])

1. So sánh chuỗi phải dùng hàm **`strcmp(s1, s2)`**: tuyệt đối KHÔNG dùng dấu "==" để so sánh (cho ra kết quả sai – còn vì sao thì hãy lên stackoverflow để biết câu trả lời).
2. Gán 2 chuỗi lẫn nhau thì dùng hàm **`strcpy(s1, s2)`** (gán s2 vào s1).
3. Trước (hoặc sau) khi nhập chuỗi (gets) thì nên có hàm **`fflush(stdin)`**.
   Vd: `fflush(stdin); gets(s1);`
   Hoặc dùng lệnh: **`cin.ignore(); //nên dùng lệnh này`**
4. Xuất chuỗi có thể dùng **`cout`** cũng được (ngoài cách dùng puts)
5. *Nhớ #include <cstring> (hoặc #include <string.h>)* – thiếu cái này là hơi mệt =)))

B. Mảng (array): quy ước gọi a là mảng *(nhớ #include <algorithm>)*

1. C++ có hàm sắp xếp **`sort:`** (để sắp tăng dần)
   `sort(a +` <vị trí bắt đầu>, `a +` <số lượng phần tử>`);`
   Ví dụ: sắp xếp a = {4, 2, 1, 5, 3} có 5 phần tử
   - Sắp tăng dần cả mảng: **`sort(a + 0, a + 5);`** // bắt đầu tại vt 0 sắp 5 phần tử liền kề -> a = {1, 2, 3, 4, 5}
   - Sắp tăng dần từ a[1] -> a[3]: **`sort(a + 1, a + 3)`** // bắt đầu tại vt 1 sắp 3 phần tử liền kề -> a = {4, 1, 2, 5, 3}

2. Để sắp giảm dần khi dùng sort thì thêm:
   `sort(a +` <vtbđ>, `a +` <slpt>, **`greater<int>());`**

3. C++ có hàm tìm kiếm giá trị k: `find`
   `find(a +` <vt bắt đầu>, `a +` <vt kết thúc>, **`<gtrị k>);`**
   => Trả về vị trí xuất hiện đầu tiên trong a của giá trị k.
   (Tìm từ vt bắt đầu đến **trước vị trí kết thúc** 1 phần tử,
   nếu **tìm không ra** sẽ trả về **vị trí kết thúc**)
   *Để lấy được vị trí, ta lấy **giá trị hàm trừ đi tên mảng** (câu này nói cho dễ nhớ thôi).*

Ví dụ: a = {4, 2, 1, 5, 3}, k = 1;

```
int vitri =find(a + 0, a + 5, 1) - a; //vitri là 2
int vttimkhongra = find(a, a + 5, 11) - a; //bằng 5
```

4. C++ có hàm đếm số lần xuất hiện của giá trị k

   `count(a +` <vtrí bắt đầu>, `a +` <slượng ptử>, <gtrị k>`);`

   Ví dụ: a = {1, 4, 6, 4, 7, 4}, k = 4

   `count(a, a + 6, 4) = 3;` //Trong mảng a, số 4 xuất hiện 3 lần

5. C++ có hàm tìm min/max trên mảng số:
   (trả về địa chỉ của phần tử lớn nhất/nhỏ nhất trong mảng)

```
max_element(a + vtbđ, a + slpt);
int a[5] = {1, 4, 6, 7, 5};
int maxA = *max_element(a, a + 5); //tìm từ a[0]->a[4], là 7
min_element(a + vtbđ, a + slpt);
int minA = *min_element(a, a + 5); //tìm từ a[0]->a[4], là 1
```

Trên đây là một số hàm cơ bản có sẵn giúp giải quyết nhanh một số bài toán cơ bản về mảng, ta có thể dùng hàm luôn mà không cần code lại :v. Nhưng điều quan trọng là lưu ý cách dùng hàm.

C. Về số:

1. C++ có sẵn hàm tính UCLN (a, b): (nhớ #include <algorithm>)

   `__gcd(a, b);` `//2 dấu shift-gạch + gcd`

2. Tìm số nhỏ hơn: `min(a, b)` (nhớ #include <algorithm>)
3. Tìm số lớn hơn: `max(a, b)` (nhớ #include <algorithm>)
4. Đổi giá trị 2 số: `swap(a, b)` (nhớ #include <algorithm>)

Mẹo cuối cùng cực kỳ hữu ích khi bạn quá hoang mang không biết mình nên #include những thư viện nào thì hãy viết ngay 1 dòng duy nhất: (chỉ include 1 mình nó thôi)

```
#include <bits/stdc++.h>
```

Sau khi viết xong thì không cần phải #include thêm bất cứ gì nữa, cứ thoải mái mà dùng.