

## ৭ সর্ট

ধরো আমার কাছে কিছু নাম্বার আছে, আমি সেগুলোকে ছোট থেকে বড়তে সাজাবো, বা উল্টো কাজটা করবো, বড় থেকে ছোটতে সাজাবো। এই কাজটাকে বলে সর্ট করা। যদি তুমি সর্ট করার নিয়ে পড়াশুনা করে ফাটাই ফেলতে চাও তাইলে এইখানে একটু তু মারো।

STL এ সর্ট করা খুব সহজ। ধরো আমার একটা ভেক্টর `v` আছে, সেটা আমি সর্ট করবো। তাহলো আমার শুধু লিখতে হবে –

```
sort( v.begin(), v.end() );
```

তাহলে সে ছোট থেকে বড় তে ভেক্টরটাকে সর্ট করে ফেলবে। এখন ধরো আমাকে যদি আরেকটু ঝামেলার কিছু করতে বলে। যেমন ধরো চাচা চৌধুরী তার মেয়ের বিয়ে দিবে, তো সে গেলো ঘটক পাখি ভাইয়ের কাছে। ঘটক পাখি ভাইয়ের কাছে একটা ছেলে মানে, তার নাম-ধাম, তার বংশ, সে কত টাকা কামায়, তার উচ্চতা কতো, আর তার ওজন কত। ছেলেটা `সি++` এ কোড করে না জাভাতে কোড করে, সেটা নিয়ে ঘটক পাখি ভাইয়ের কোনই মাথা ব্যাথা নাই। তো সে করলো কি চাচা চৌধুরীকে শুধু এই ক্যাটা ডাটাই সাপ্লাই দিলো কয়েকটা বস্তা ভরে। এখন চাচা চৌধুরী পাড়ার প্যান্ট টিলা মাস্তানের কাছ থেকে শুনলো তুমি একটা বস প্রোগ্রামার, তো সে এসে তোমাকে বলল, "বাবাজি! আমাকে একটা সফটওয়্যার বানিয়ে দাও, যেটা আমার ডাটাগুলোকে সাজাবে"।

বেশ তো, এখন আমার ডাটাটা হচ্ছে এরকম - (চাচা চৌধুরী আবার বংশ নিয়ে মাথা ঘামায় না)

```
struct data {  
    char name[100];  
  
    int height, weight;  
  
    long long income;  
};
```

চাচা চৌধুরী যেটা নিয়ে মাথা ঘামায় সেটা হলো পোলার কত টাকা কামাই। যদি দুইটা পোলার সমান কামাই হয়, তাইলে যেই পোলার হাইট ভালো, সেই পোলা লিস্ট আগে থাকবে। আর যদি দুই পোলার হাইট সমান হয় তাইলে যেই পোলার ওজন কম, সেই পোলা আগে থাকবে। আর যদি দুই পোলার ওজন সমান হয়, তাইলে যেই পোলার নাম ছোট সেই পোলা আগে থাকবে।

এখন তোমাকে এই অনুযায়ী সর্ট করে দিতে হবে। আর তুমি যদি বেশি হাংকি পাংকি করো, তাইলে প্যান্ট টিলা মাস্তান এসে তোমাকে সাইজ করে দিবে।

এই কাজটা দুই ভাবে করা যায়। সবচে সহজটা হলো একটা কম্পায়ার ফাংশন লিখে।

```
bool compare( data a, data b ) {  
  
    if( a.income == b.income ) {
```

```

if( a.height == b.height ) {

    if( a.weight == b.weight )

        return strlen( a.name ) < strlen( b.name );

    else return a.weight < b.weight;

} else return a.height > b.height;

} else return a.income > b.income;

}

```

এই ফাংশনটা গ্লোবালি ডিক্লেয়ার করে যেখানে তুমি সর্ট করতে চাও সেখানে লিখতে হবে।

```
sort( v.begin(), v.end(), compare );
```

কম্পায়ার ফাংশনটা রিটার্ন করবে a কি b এর আগে বসবে কি না। আর কিছু না।

সর্ট করার অন্য পথটা হচ্ছে অপারেটর ওভারলোড করে। ধরো, আমরা যখন বলি  $2 < 3$  আমরা বুঝে নেই যে ২ হচ্ছে ৩ এর ছোট - মানের দিক দিয়ে। এখন একটা স্ট্রাকচার কখন অন্য আরেকটা স্ট্রাকচারের চেয়ে ছোট হবে? এই জিনিসটা তোমার প্রোগ্রামে ডিফাইন করে দিতে হবে। এখানে খেয়াল করো, ছোট হবার মানে বোঝাচ্ছে সে লিস্টে আগে থাকবে।

আমি যদি একই কাজটা অপারেটর ওভারলোড দিয়ে করতে চাই, সেটা এরকম হবে।

```

struct data {

    char name[100];

    int height, weight;

    long long income;

    bool operator < ( const data& b ) const {

        if( income == b.income ) {

            if( height == b.height ) {

                if( weight == b.weight )

                    return strlen( name ) < strlen( b.name );

```

```
        else return weight < b.weight;

    }else return height > b.height;

    }else return income > b.income;

}

};
```

এখানে কিন্তু আমি এই ডাটাটাকেই অন্য আরেকটা ডাটা b এর সাথে তুলনা করছি, সেজন্য আমার আগেরটার মতো a কে লাগছে না।

আর আমার সর্ট এর কমান্ড লিখতে হচ্ছে এইভাবে।

```
sort( v.begin(), v.end() );
```

তোমার যদি ভেক্টর ব্যবহার করতে আপত্তি থাকে, ধরো ভেক্টর দেখলেই হাঁচি আসা শুরু করে, নাক চুলকায় কিংবা এধরণের কিছু, তুমি সাধারণ অ্যারেই ব্যবহার করতে পারো।

ধরো সেক্ষেত্রে অ্যারেটা হবে এরকম -

```
data array[100];
```

```
sort( array, array + n );
```

যেখানে n হচ্ছে অ্যারেতে কতগুলো ডাটাকে তুমি সর্ট করতে চাও।

তুমি যদি 3 নাম্বার (0 based)থেকে 10 নাম্বার পর্যন্ত সর্ট করতে চাও লিখো

```
sort( array+3, array+11 );
```

# Tanver Ahmed Likhon