STRUCT COMPARATORS

PREREQUISITES:

1) Language Constructs - struct
2) Sorting
3) STL - sort()
4) User Defined Functions

MOTIVATIONAL PROBLEM:

Let us take the same problem as we took while learning Language Construct- structs.

We have a school of N students. And we have to accept the following details for each student: Name of the student, Roll number of the student, class of the student, Marks in Physics , Chemistry and Computer Science. We are also asked to calculate and store the average marks for each student.

I hope you remember how we defined a ‘student’ data structure(using struct command) and we had an array of N elements where each element of the array was a ‘student’ data structure we defined.
If you do not know how to organise data this way,
I would highly recommend you to go through Language Construct - structs tutorial.

NOW WHAT IF I WANTED THE ARRAY SORTED ACCORDING TO SOME DETAIL(S)?

Suppose after calculating the average I ask you to sort the student array according to the increasing order of physics marks.
How would you do this?
Can you use the STL sort function you learnt?
You may try writing something like this: sort(<array name> , <array name>+size of the array), as you learnt.
But do you think this will work?
Now, what if I ask you to sort the array according to increasing order of average marks? What would you have written?
Same thing?
Now my question to you is how will the computer know whether you want to sort it according to increasing order of Physics marks, or Chemistry marks or Average Marks.
So clearly writing sort(<array name> , <array name>+size of the array) will not work. :(

DO WE WRITE MERGE SORT AGAIN ON OUR OWN?

So does that mean that I have to write my own merge sort function?
But that is too much of work! (Who wants to write that merge sort in a 2 hour programming contest and waste the time).

WE CAN USE STL sort() ! BUT HOW?

It turns out that there is indeed a way to use the STL sort() but I need to specify how do I want the computer to sort, whether it is according to Physics marks or Chemistry marks or Computer marks or Average marks, whether it is in ascending order or descending order.
How do we do this in code?
Instead of writing: sort(<array name> , <array name> + size of the array) we write:
sort(<array name> , <array name> + size of the array, <name of comparator>).

<u>WHAT'S A COMPARATOR?</u>

You can think of a comparator as a user defined function with a bool return type. It takes as arguments, two elements. These elements can be anything. It can be integers, strings or even our own data structure student.
Inside this function we specify how do we want the elements to be sorted.

The following code is for the comparator(comp) which will help us sort the array A which contains the details of N students according to the Physics marks:
****Note that if we want to sort an array which has 'student' data structure as its elements using a comparator, we must pass two 'student' data structures as the two arguments of the comparator function(comp()). If instead, we want to sort an array which has int as its elements, using a comparator, the two arguments of the comparator function must be ints.

```
bool comp(student X, student Y)
{
        return X.phy<Y.phy;
}
```

Notice the < sign, if we want to sort in increasing order of some details.
Okay. Now what will the comparator look like if I want to sort according to decreasing order of chemistry marks?

Soln.:

```
bool comp(student X, student Y)
{
        return X.ch>Y.ch;
}
```

Notice the > sign, if we want to sort in decreasing order of some details.

<u>GOLDEN RULE!</u>

We can make it a formula that if we are asked to sort according to increasing order use the < sign.
If we are asked to sort according to decreasing order we use the > sign.
Make sure you do not change the order of the first and second parameters.
In that case the symbols(>,<) would be reversed!

<u>BUT, HOW DOES THIS ACTUALLY WORK?</u>

I will not go into the details of this. I will give an informal description of how the sort function sorts according to the specified comparator.
Let us consider the first comparator we wrote which sorts according to increasing order of Physics marks.
This comp function(comparator) returns true if student X(the first parameter) is supposed to be found before student Y(the second parameter) in the array A.
So if X.phy < Y.phy, the comp() would return true.
And this is indeed correct because X is supposed to be before Y in the array since we are sorting according to increasing order of Physics marks of students.
Depending on whether comp() returns true or false for two elements, the sort function changes the position of those two elements in the array.

It is beyond the scope of this tutorial to go into further details.

EXERCISE:

1. Now let's do a quick exercise:
   Can you write a comparator that sorts an int array in descending order of values. Call the sort() from main() with appropriate syntax.
   Link: https://ideone.com/Kb5ms6
2. Now can you write only the comparator to sort the student array in descending order of average marks and break ties according to decreasing order of Physics marks. In other words a student with higher average is placed before a student with lower average. If two students have the same average the one who secured higher Physics marks will be placed above the one with lower Physics marks. You may assume that the student data structure has the following variables:
   name
   roll
   standard
   ph
   ch
   comp
   avg
   Write only the comparator.
   Soln.: https://ideone.com/cVT9Cx
   This type of problem is very common in Competitive Programming world. Practice this a couple of times before you get comfortable with it.

CONCLUSION

Now you completely understand how to sort data structures according to some feature using the STL sort(). You also know how to create your own data structure. Congratulations!