C++ Standard Template library

Every wise programmer knows "**to not re-invent the wheel**"
Therefore, we use a standard library for different data structures, to save our time in contest
The main components of STL are :

- Container
- Algorithms
- Iterators
- Functions
- Utility library

Containers

There are few basic container templates:

- Vector
- Deque
- Stack
- Priority-Queue
- Set
- Multi-set
- Map
- Multi-Map

Vector

Vectors are used as dynamic arrays, there are other properties as well. Let's not get there and learn different operations on vectors.

1. Creating a vector

   ```
   vector<data-type> vector_name;
   ```

   Example

   ```
   vector<int> v;
   ```

2. Inserting an element in vector

   ```
   vector_name.push_back(data);
   ```

Example

```
int data = 10;
v.push_back(data);
```

Deque

Some good stuff, but not going there yet. We will come back to it

Forward List

Again Some good stuff, but not going there yet. We will come back to it

Set

Sets can be said container which is store data in sorted manner

- Very important property of set is that it will not store same data

But actually set implementation behind the scene is by using an abstract data structure, which says its documenation, but I believe it is ought to be Red-Black Tree.

Useful operations on set

1. Creating a set

   ```
   set<data-type> set_name;
   ```

   Example

   ```
   set<int> s;
   ```

2. Inserting an element in set

   ```
   set_name.insert(data);
   ```

   Example

   ```
   int data = 10;
   s.insert(data);
   ```

Map

Maps are good

- Very important property of map is that it will not store same key again

Map implemenation behind the scene is a Red-Black Tree so insertion in map is $O(log(n))$ which is same for accessing an element

Useful operations on set

1. Creating a map

```
map<data-type-for-key, data-type-for-key-data> map_name;
```

Example

```
map<int,int> m;
```

2. Inserting an element in set

```
map_name[key] = data;
```

Example

```
int data = 10;
int key = 15;
m[key] = data;
```