**[L1]Class - 8 ( Almost everything about STL Set ) [Bangla]**

[**https://www.youtube.com/watch?v=cH\_w5xNGEco&list=PLoa\_roVVsxA1CJu4DsOljb9d7FJkMCynM&index=8&pbjreload=101**](https://www.youtube.com/watch?v=cH_w5xNGEco&list=PLoa_roVVsxA1CJu4DsOljb9d7FJkMCynM&index=8&pbjreload=101)

**Video link :**

[**https://youtu.be/cH\_w5xNGEco**](https://youtu.be/cH_w5xNGEco)

**Discussed topics :**

1) STL Set

**Code :**

<https://pastebin.com/d1SFD8TK>

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

using namespace std;

*#define endl '\n'*

*#define optimeze() ios\_base::sync\_with\_stdio(0);cin.tie(0);cout.tie(0);*

int main()

{

optimeze();

set<int>s={1,1,3,3,2,2};

cout<<s.size()<<endl;//3

**for**(auto u:s) cout<<u<<" ";//1 2 3

cout<<endl<<endl;

set<int>s1={1,1,3,3,2,2};

set<int>::iterator it;

**for**(it=s1.begin();it!=s1.end();it++) cout<<\*it<<" ";//1 2 3

cout<<endl<<endl;

set<int>s2={1,1,3,3,2,2};

s2.clear();

cout<<s2.empty()<<endl<<endl;//1

set<int>s3;

s3.insert(2);

s3.insert(2);

s3.insert(3);

s3.insert(1);

cout<<s3.size()<<endl;//3

**for**(auto u:s3) cout<<u<<" ";//1 2 3

cout<<endl<<endl;

set<int>s4;

s4.insert(2);

s4.insert(2);

s4.insert(3);

s4.insert(1);

cout<<s4.count(2)<<endl;//1

cout<<\*s4.begin()<<endl;//1

cout<<\*(--s4.end())<<endl;//3

cout<<\*(s4.rbegin())<<endl<<endl;//3

s4.erase(2);

cout<<s4.size()<<endl;//2

**for**(auto u:s4) cout<<u<<" ";//1 3

cout<<endl<<endl;

set<int>s5;

s5.insert(1);

s5.insert(2);

s5.insert(3);

s5.insert(4);

s5.insert(5);

s5.insert(6);

s5.erase(s5.begin());//1

s5.erase(--s5.end());//6

cout<<s5.size()<<endl;

**for**(auto u:s5) cout<<u<<" ";//2 3 4 5

cout<<endl<<endl;

set<string>s6;

s6.insert("shariar");

s6.insert("proma");

s6.insert("momo");

s6.insert("sobuj");

s6.insert("prety");

s6.insert("nobel");

s6.insert("shariar");

s6.insert("proma");

s6.insert("momo");

s6.insert("sobuj");

s6.insert("prety");

s6.insert("nobel");

cout<<s6.size()<<endl;//6

**for**(auto u:s6) cout<<u<<endl;

cout<<endl<<endl;

/\*

momo

nobel

prety

proma

shariar

sobuj

\*/

set<pair<int,int>>p;

p.insert({2,3});

p.insert({4,1});

p.insert({4,1});

p.insert({2,1});

p.insert({4,3});

p.insert({5,1});

p.insert({5,1});

p.insert({5,4});

cout<<p.size()<<endl;//6

**for**(auto u:p) cout<<u.first<<" "<<u.second<<endl;

cout<<endl<<endl;

/\*

2 1

2 3

4 1

4 3

5 1

5 4

\*/

set<int,greater<int>>g;

g.insert(1);

g.insert(1);

g.insert(2);

g.insert(3);

cout<<g.size()<<endl;//3

**for**(auto u:g) cout<<u<<" ";//3 2 1

cout<<endl<<endl;

set<string,greater<string>>a;

a.insert("shariar");

a.insert("proma");

a.insert("momo");

a.insert("sobuj");

a.insert("prety");

a.insert("nobel");

a.insert("shariar");

a.insert("proma");

a.insert("momo");

a.insert("sobuj");

a.insert("prety");

a.insert("nobel");

cout<<s.size()<<endl;//6

**for**(auto u:a) cout<<u<<endl;

cout<<endl<<endl;

/\*

sobuj

shariar

proma

prety

nobel

momo

\*/

set<pair<int,int>,greater<pair<int,int>>>p1;

p1.insert({2,3});

p1.insert({4,1});

p1.insert({4,1});

p1.insert({2,1});

p1.insert({4,3});

p1.insert({5,1});

p1.insert({5,1});

p1.insert({5,4});

cout<<p1.size()<<endl;//6

**for**(auto u:p1) cout<<u.first<<" "<<u.second<<endl;

cout<<endl<<endl;

/\*

5 4

5 1

4 3

4 1

2 3

2 1

\*/

**return** 0;

}

"C:\Users\Md.Hamid Hosen\CLionProjects\untitled\cmake-build-debug\untitled.exe"

3

1 2 3

1 2 3

1

3

1 2 3

1

1

3

3

2

1 3

4

2 3 4 5

6

momo

nobel

prety

proma

shariar

sobuj

6

2 1

2 3

4 1

4 3

5 1

5 4

3

3 2 1

3

sobuj

shariar

proma

prety

nobel

momo

6

5 4

5 1

4 3

4 1

2 3

2 1

Process finished **with** exit code 0

In [ ]:

/// \*\*\* --- ||| In the name of ALLAH ||| --- \*\*\* ///

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

using namespace std;

int main()

{

/// Declaration of set of integers

set<int> s = { 1, 1, 2, 1, 3 };

/// Printing size **and** elements

cout << s.size() << endl;

**for** ( auto u : s ) cout << u << " "; /// 1 2 3

cout << endl;

set<int>:: iterator it;

**for** ( it = s.begin(); it != s.end(); it++ ) cout << \*it << " "; /// 1 2 3

cout << endl;

/// clearing **and** checking **is** empty set

s.clear();

cout << s.empty() << endl; /// 1

/// inserting **in** set

s.insert ( 1 );

s.insert ( 1 );

s.insert ( 1 );

s.insert ( 2 );

s.insert ( 1 );

s.insert ( 3 );

cout << s.size() << endl; /// 3

**for** ( auto u : s ) cout << u << " "; /// 1 2 3

cout << endl;

/// checking **is** specific element **is** **in** a set

cout << s.count ( 2 ) << endl; /// 1

cout << s.count ( 4 ) << endl; /// 0

/// Front element **in** set

cout << \*s.begin() << endl; /// 1

/// Last element **in** set

cout << \*(--s.end()) << endl; /// 3

cout << \*s.rbegin() << endl; /// 3

/// Erase an element;

s = { 1, 2, 3, 4, 5, 6 };

s.erase ( 2 );

cout << s.size() << endl; /// 5

**for** ( auto u : s ) cout << u << " "; /// 1 3 4 5 6

cout << endl;

s.erase ( 10 );

cout << s.size() << endl; /// 5

**for** ( auto u : s ) cout << u << " "; /// 1 3 4 5 6

cout << endl;

/// Erasing front element **in** set

s = { 1, 2, 3, 4, 5, 6 };

s.erase ( s.begin() );

cout << s.size() << endl; /// 5

**for** ( auto u : s ) cout << u << " "; /// 2 3 4 5 6

cout << endl;

/// Erasing back element **in** set

s = { 1, 2, 3, 4, 5, 6 };

s.erase ( --s.end() );

cout << s.size() << endl; /// 5

**for** ( auto u : s ) cout << u << " "; /// 1 2 3 4 5

cout << endl;

/// set of pair

set<pair<int, int>> s1;

s1.insert ( { 1, 2 } );

s1.insert ( { 1, 2 } );

s1.insert ( { 4, 2 } );

s1.insert ( { 4, 3 } );

s1.insert ( { 2, 2 } );

s1.insert ( { 2, 1 } );

/\*\*

Output :

5

1 2

2 1

2 2

4 2

4 3

\*/

cout << s1.size() << endl;

**for** ( auto u : s1 ) cout << u.first << " " << u.second << endl;

/// Set of string

set<string> s2;

s2.insert ( "momo" );

s2.insert ( "momo" );

s2.insert ( "prety" );

s2.insert ( "prety" );

s2.insert ( "shahriar" );

s2.insert ( "nobel" );

s2.insert ( "sharif" );

s2.insert ( "proma" );

cout << s2.size() << endl;

**for** ( auto u : s2 ) cout << u << endl;

/\*\*

Output :

6

momo

nobel

prety

proma

shahriar

sharif

\*/

/// set **in** discanding order

set<int, greater<int>> s3 = {3, 4, 1, 2};

**for** ( auto u : s3 ) cout << u << " "; /// 4 3 2 1

cout << endl;

set<string, greater<string>> s4;

s4.insert ( "momo" );

s4.insert ( "momo" );

s4.insert ( "prety" );

s4.insert ( "prety" );

s4.insert ( "shahriar" );

s4.insert ( "nobel" );

s4.insert ( "sharif" );

s4.insert ( "proma" );

cout << s4.size() << endl;

**for** ( auto u : s4 ) cout << u << endl;

/\*\*

Output :

6

sharif

shahriar

proma

prety

nobel

momo

\*/

set<pair<int, int>, greater<pair<int,int>>> s5;

s5.insert ( { 1, 2 } );

s5.insert ( { 1, 2 } );

s5.insert ( { 4, 2 } );

s5.insert ( { 4, 3 } );

s5.insert ( { 2, 2 } );

s5.insert ( { 2, 1 } );

cout << s5.size() << endl;

**for** ( auto u : s5 ) cout << u.first << " " << u.second << endl;

/\*\*

Output :

5

4 3

4 2

2 2

2 1

1 2

\*/

**return** 0;

}

**STL set lower\_bound && upper\_bound:**

#include**<bits/stdc++.h>  
using namespace** std;  
#define **optimize**() ios\_base::sync\_with\_stdio(0);cin.tie(0);cout.tie(0);  
#define **endl '\n'  
int** main()  
{  
 **optimize**();  
  
 set<**int**>s={1,2,3,4,5};  
 cout<<\*lower\_bound(s.begin(),s.end(),2);*//2* cout<<\*upper\_bound(s.begin(),s.end(),2);*//3* **return** 0;  
}