**STL Part**

**Almost Everything about STL Vector**

**vector<int> v;  
  
v.push\_back( 1 );  
v.push\_back( 2 );  
v.push\_back( 3 );  
  
cout << v[0] << " " << v[1] << " " << v[2] << endl; */// 1 2 3*v[1] = 3;  
cout << v[0] << " " << v[1] << " " << v[2] << endl; */// 1 3 3*cout << v.size() << endl; */// 3*for ( int i = 0; i < v.size(); i++ ) cout << v[i] << " "; */// 1 3 3*cout << endl;  
  
vector <int> v1 = { 2, 3, 4 };  
  
cout << v1.size() << endl; */// 3*for ( int i = 0; i < v1.size(); i++ ) cout << v1[i] << " "; */// 2 3 4*cout << endl;  
  
v.clear();  
cout << v.size() << endl; */// 0*cout << v.empty() << endl; */// 1*cout << v1.empty() << endl; */// 0*v1.resize(5);  
cout << v1.size() << endl; */// 5*for ( int i = 0; i < v1.size(); i++ ) cout << v1[i] << " "; */// 2 3 4 0 0*cout << endl;  
  
vector<int> a(5);  
  
cout << a.size() << endl; */// 5*for ( int i = 0; i < a.size(); i++ ) cout << a[i] << " "; */// 0 0 0 0 0*cout << endl;  
  
a = v1;  
  
for ( auto u : a ) cout << u << " "; */// 2 3 4 0 0*cout << endl;  
  
vector<int>::iterator it;  
for ( it = a.begin(); it != a.end(); it++ ) cout << \*it << " "; */// 2 3 4 0 0*cout << endl;  
  
a = { 3, 4, 5, 1, 2 };  
  
sort ( a.begin(), a.end() ); *///O(n\*log2(n))*for ( auto u : a ) cout << u << " "; */// 1 2 3 4 5*cout << endl;  
  
sort ( a.rbegin(), a.rend() );  
  
for ( auto u : a ) cout << u << " "; */// 5 4 3 2 1*cout << endl;  
  
  
a = { 3, 4, 5, 1, 2 };  
sort ( a.begin(), a.end(), greater<int>() );  
  
for ( auto u : a ) cout << u << " "; */// 5 4 3 2 1*cout << endl;  
  
a = { 3, 4, 5, 1, 2 };  
  
reverse( a.begin(), a.end() );  
  
for ( auto u : a ) cout << u << " "; */// 2 1 5 4 3*cout << endl;  
  
cout << a.back() << endl; */// 3*a.pop\_back(); */// O(1) complexity.*cout << a.back() << endl; */// 4*a = { 3, 4, 5, 1, 2 };  
cout << \*a.begin() << endl; */// 3*a.erase( a.begin() ); */// O(n) complexity.*for ( auto u : a ) cout << u << " "; */// 4 5 1 2*cout << endl;  
  
a.erase( a.begin()+2 );  
for ( auto u : a ) cout << u << " "; */// 4 5 2*cout << endl;  
  
a = { 1, 1, 2, 2, 2, 3, 3 };  
unique( a.begin(), a.end() );  
  
for ( auto u : a ) cout << u << " "; */// 1 2 3 2 2 3 3*cout << endl;  
  
a = { 1, 1, 2, 2, 2, 3, 3 };  
int n = unique( a.begin(), a.end() ) - a.begin();  
  
cout << n << endl; */// 3*for ( int i = 0; i < n; i++ ) cout << a[i] << " "; */// 1 2 3*cout << endl;  
  
a = { 2, 3, 1, 5 };  
cout << max\_element( a.begin(), a.end() ) - a.begin() << endl; */// 3*cout << \*max\_element( a.begin(), a.end() ) << endl; */// 5*cout<<\*max\_element(a.begin(),a.end())<<endl;*//5*cout<<\*min\_element(a.begin(),a.end())<<endl;*//1***

**//Again**

**vector<int>v={2,3,5,5,7,7,1};  
  
sort(v.begin(),v.end());*//O(nlog2(n))*int size=unique(v.begin(),v.end())-v.begin();*///O(n)*cout<<"Unique Size:"<<size<<endl;  
for(int i=0;i<size;i++){  
 cout<<v[i]<<' ';  
}  
cout<<endl;  
  
for(auto u:v)cout<<u<<" ";  
cout<<endl;  
  
*/\*  
 \* Unique Size:5  
 1 2 3 5 7  
 1 2 3 5 7 7 7  
 \*/*vector<int>v={2,3,5,5,7,7,1};  
  
cout<<"Maximum value print::"<<endl;  
cout<<\*max\_element(v.begin(),v.end())<<endl;  
*//or*vector<int>::iterator it=max\_element(v.begin(),v.end());  
cout<<\*it<<endl;  
  
vector<int>::iterator it1=max\_element(v.begin()+1,v.begin()+4);  
cout<<\*it1<<endl;  
  
cout<<"Index=";  
int n=max\_element(v.begin(),v.end())-v.begin();  
cout<<n<<endl;  
  
*/\*  
 \* Maximum value print::  
 7  
 7  
 5  
 Index=4  
 \*/***

**2D array:**

**int row=4,col=5,value=5;  
vector<vector<int>>m;  
  
m.resize(row,vector<int>(row,value));  
  
for(int i=0;i<row;i++){  
 for(int j=0;j<m[i].size();j++) {  
 cout<<m[i][j]<<" ";  
 }  
 cout<<endl;  
}**

**Output:**

5 5 5 5

5 5 5 5

5 5 5 5

5 5 5 5

**Copy one vector to another vector:**

**vector<int>v,v1;  
v.push\_back(1);  
v.push\_back(2);  
v.push\_back(3);  
  
v1=v;  
  
for(auto u:v1){  
 cout<<u<<" ";  
}**

**Output:**

**1 2 3**

**Vector assign ():**

**vector<int>v;  
v.assign(3,4);  
for(auto u:v){  
 cout<<u<<" ";  
}**

**Output:**

**4 4 4**

**The syntax for assigning values from an array or list:**

**vector<int>v;  
int a[]={1,2,3};  
v.assign(a,a+3);  
for(auto u:v){  
 cout<<u<<" ";  
}**

**Output:**

**1 2 3**

**Syntax for assigning values with initialize list:**

**vector<int>v;  
v.assign({ 1, 2, 3 });  
for(auto u:v){  
 cout<<u<<" ";  
}**

**Output:**

**1 2 3**

**The syntax for modifying values from a vector**:

**vector<int> v;  
v.assign(7, 100);  
for(auto u:v){  
 cout<<u<<" ";  
}  
cout<<endl;  
v.assign(v.begin(), v.begin() + 3);  
for(auto u:v){  
 cout<<u<<" ";  
}**

**Output:**

100 100 100 100 100 100 100

100 100 100

**Insert function: vector\_name.insert(pos,value)**

**vector<int> v = { 10, 20, 30, 40 };  
v.insert(v.begin()+2,1000);*//vector\_name.insert(pos,value)*v.insert(v.begin(),2000);  
for(auto u:v){  
 cout<<u<<" ";  
}**

**Output:**

**2000 10 20 1000 30 40**

**Insert function : vector\_name.insert(position, size, val)**

**vector<int> v = { 10, 20, 30, 40 };  
v.insert(v.begin(),2,2000);*//vector\_name.insert(position, size, val)*for(auto u:v){  
 cout<<u<<" ";  
}**

**Insert function**: *vector\_name.insert(position, iterator1, iterator2)*

**vector<int> v1 = { 10, 20, 30, 40 };  
vector<int> v2;  
  
v2.insert(v2.begin(),v1.begin(),v1.end());*//vector\_name.insert(position, iterator1, iterator2)  
//10 20 30 40*v2.insert(v2.begin(),v1.begin(),v1.begin()+2);*//10 20*for(auto u:v2){  
 cout<<u<<" ";  
}**

**Vector Problem:**

1. Code force=381A(Sereja and Dima)
2. LeetCode=852( Peak Index in a Mountain Array)
3. LeetCode= 1502(Can Make Arithmetic Progression From Sequence)
4. Hackerearth=(Distinct Count)

# Almost everything about STL String

***/// Declare string*string s;  
  
*/// Assign string*s = "abcdf";  
  
*/// Printing size of string*cout << s.size() << endl; */// 5  
  
/// Printing string*cout << s << endl; */// abcdf  
  
/// Pushing char back to a string*s += 'b';  
s += 'c';  
cout << s << endl; */// abcdfbc  
  
/// Taking input string*cin >> s;  
cout << s << endl;  
  
s = "asdfgg";  
  
*/// Checking is a string empty or not*string s1;  
cout << s.empty() << endl; */// 0*cout << s1.empty() << endl; */// 1  
  
/// Assigning an string in another string variable*s1 = s;  
s.clear();  
  
cout << s.empty() << endl; */// 1*cout << s1.empty() << endl; */// 0  
  
/// assigning 'k' in 0-th index*s = "asdfg";  
s[0] = 'k';  
cout << s << endl; *///kasdfg;*s = "abc";  
s1 = "def";  
  
*/// String concatenation*string tmp = s + s1;  
cout << tmp << endl; */// abcdef  
  
/// String iterator*string::iterator it;  
for ( it = s.begin(); it != s.end(); it++ ) cout << \*it; */// abc*cout << endl;  
  
*/// For each loop*for ( auto c : s ) cout << c; */// abc*cout << endl;  
  
  
s = "asd";  
tmp = s;  
  
*/// Comparing two strings*if ( tmp == s ) cout << "Yes Match\n";  
else "No Match\n";  
  
*/// String reverse and checking is a string is palindrome or not*s = "asddsa";  
tmp = s;  
reverse( tmp.begin(), tmp.end() );  
  
if ( tmp == s ) cout << "Yes Palindrome" << endl;  
else cout << "Not Palindrome" << endl;  
  
*/// String sorting in non-decreasing order*s = "gfds";  
sort ( s.begin(), s.end() );  
cout << s << endl; */// dfgs  
  
/// String sorting in non-increasing order*sort ( s.rbegin(), s.rend() );  
cout << s << endl; */// sgfd  
  
/// Getting all unique elements of a string. Be care full, string should be sorted.*s = "aaadddsss";  
int n = unique( s.begin(), s.end() ) - s.begin();  
for ( int i = 0; i < n; i++ ) cout << s[i];*/// ads*cout << endl;  
  
*/// Getting maximum element of string*cout << \*max\_element( s.begin(), s.end() ) << endl; */// s  
/// Getting minimum element of string*cout << \*min\_element( s.begin(), s.end() ) << endl; */// a  
  
/// When we want to take input with space  
/// input : Muhammad Shahriar Alam*char c;  
cin >> c;  
getline( cin, s );  
s = c + s;  
  
cout << s << endl; */// Muhammad Shahriar Alam  
  
/// If we need to sort some string on lexicographical order :*vector<string> v;  
v.push\_back( "Muhammad" );  
v.push\_back( "Nova" );  
v.push\_back( "Maslenia Mubarrat" );  
v.push\_back( "CPS Academy" );  
v.push\_back( "Rashedul Alam Anik" );  
v.push\_back( "Farhan sadik Sakib" );  
v.push\_back( "Gazi Mohaimin Iqbal" );  
  
sort ( v.begin(), v.end() );  
for ( auto u : v ) cout << u << endl;  
  
*/\*\*  
  
Out put :  
  
CPS Academy  
Farhan sadik Sakib  
Gazi Mohaimin Iqbal  
Maslenia Mubarrat  
Muhammad  
Nova  
Rashedul Alam Anik  
  
\*/*s = "asdf";  
  
s.pop\_back(); */// removes last char of string*cout << s.back() << endl; */// print last char of string*v.clear();  
  
v = { "Shahriar", "Shahriar", "Momo", "Momo", "Sharif", "Sharif" };  
int Sz = unique ( v.begin(), v.end() ) - v.begin();  
  
  
cout << Sz << endl; */// Number of unique strings in vector v;*for ( int i = 0; i < Sz; i++ ) cout << v[i] << endl; */// Prints all unique strings in vector v  
  
/// Converting int to string*int a = 123;  
s = to\_string (a);  
cout << s << endl; */// 123*s[0] = '3';  
cout << s << endl; */// 323  
  
/// Converting string to integer*s = "123";  
a = stoi ( s );  
cout << a << endl; */// 123*a++;  
cout << a << endl; */// 124;  
  
/// Deleting a substring from string*s = "ShaKAKAhriar";  
  
s.erase ( s.begin()+3, s.begin()+7 ); */// erase substring "KAKA" from string s*cout << s << endl;  
  
  
*/// Copying a substring of a string to a string*tmp = "Gagha Alam Gadha";  
s = "Shahriar ";  
  
copy ( tmp.begin()+6, tmp.begin()+10, back\_inserter ( s ) ); */// copying "Alam substring to string s back.*cout << s << endl; */// Shahriar Alam  
  
/// Erasing all occurrence of a specific char from string.*s = "aaassdddaaasdd";  
s.erase ( remove ( s.begin(), s.end(), 'a' ), s.end() ); */// removes all 'a' from s*cout << s << endl;  
  
*/// Checking is a string is substring of another string in O(n\*m)*s = "ashshasdakks";  
  
if ( s.find( "asd" ) != -1 ) cout << "Substring found";*//"Substring found"*else cout << "Not found";**

**String Problem:**

1. Codeforce=118A(String Task)
2. CodeForce=112A(Petya and String)
3. CodeForce=78A(Haiku)
4. Hackerearth=(The Palindrome)
5. Hackerearth=(LexoGeek)
6. Geeksforgeeks=Sum of Digit is Pallindrome or not

# Almost everything about STL pair

**#include<bits/stdc++.h>  
using namespace std;  
#define endl '\n'  
#define optimize() ios\_base::sync\_with\_stdio(0);cin.tie(0);cout.tie(0);  
int main()  
{  
 optimize();  
  
 pair<int,int>p;  
 p.first=2;  
 p.second=5;  
  
 cout<<p.first<<endl;*//2* cout<<p.second<<endl<<endl;*//5* pair<string,int>q;  
 q.first="Hamid";  
 q.second=5;  
  
 cout<<q.first<<endl;*//Hamid* cout<<q.second<<endl<<endl;*//5* pair<string,vector<int>>x;  
 x.first="hamid";  
 x.second= {1,2,3};  
  
 cout<<x.first<<endl;*//hamid* for (auto u:x.second)cout<<u<<" ";*//1 2 3* cout<<endl;  
  
 pair<int,int>y;  
 y=make\_pair(2,3);  
  
 cout<<y.first<<" "<<y.second<<endl<<endl;*//2 3* pair<int,int>a;  
 a= {2,3};  
  
 cout<<a.first<<" "<<a.second<<endl<<endl;*//2 3* pair<string,vector<int>>s;  
 *//s={"hamid",{3,4,5}};* s.first="hamid",s.second= {3,4,5};  
  
 cout<<s.second.size()<<endl;*//3* cout<<s.first.size()<<endl;*//5* cout<<s.first<<endl;*//hamid* for(auto u:s.second)cout<<u<<" ";*//3 4 5* cout<<endl<<endl;  
  
 pair<int,int>p1,p2,p3,p4;  
 p1= {2,3},p2= {3,2};  
  
 if(p1<p2)cout<<"Yes"<<endl<<endl;*//yes* else cout<<"No"<<endl<<endl;  
  
 p3=min(p1,p2);  
 cout<<p3.first<<" "<<p3.second<<endl<<endl;*//2 3* p4=max(p1,p2);  
 cout<<p4.first<<" "<<p4.second<<endl<<endl;*//3 2  
  
 //pair sort* vector<pair<int,int>>v;  
  
 v.push\_back({6,5});  
 *//v.push\_back(make\_pair(6,5));* v.push\_back({2,3});  
 v.push\_back({4,5});  
 v.push\_back({6,1});  
 v.push\_back({1,9});  
  
 *//sort(v.begin(),v.end());  
 //sort(v.rbegin(),v.rend());* for(auto u:v) cout<<u.first<<" "<<u.second<<endl;  
 cout<<endl<<endl;  
 */\*  
 6 5  
 2 3  
 4 5  
 6 1  
 1 9  
 \*/* pair<int,int> c[]= {{6,5},{2,3},{4,5},{6,1},{1,9}};  
 sort(c,c+5);  
 *//sort(c,c+5,greater<int>());* for(auto u:c)cout<<u.first<<" "<<u.second<<endl;  
 cout<<endl<<endl;  
 */\*  
 1 9  
 2 3  
 4 5  
 6 1  
 6 5  
 \*/* vector<pair<string,int>> d;  
  
 d.push\_back({"hamid",21});  
 d.push\_back({"momo",13});  
 d.push\_back({"hamid",34});  
 d.push\_back({"hosi",35});  
 d.push\_back({"hamid",34});  
  
 sort(v.begin(),v.end());  
  
 for(auto u:d) cout<<u.first<<" "<<u.second<<endl;  
 cout<<endl<<endl;  
 */\*  
 hamid 21  
 momo 13  
 hamid 34  
 hosi 35  
 hamid 34  
 \*/* int array[]={1,2,4,5};  
 for(auto u:array) cout<<u<<" ";*//1 2 4 5* cout<<endl<<endl;  
  
 vector<pair<string,int>> e;  
  
 e.push\_back({"hamid",21});  
 e.push\_back({"hamid",21});  
 e.push\_back({"momo",13});  
 e.push\_back({"momo",13});  
 e.push\_back({"hamid",34});  
 e.push\_back({"hosi",35});  
 e.push\_back({"hamid",34});  
  
 sort(e.begin(),e.end());  
  
 int size=unique(e.begin(),e.end())-e.begin();  
  
 for(int i=0;i<size;i++) cout<<e[i].first<<" "<<e[i].second<<endl;  
 cout<<endl<<endl;  
 */\*  
 hamid 21  
 hamid 34  
 hosi 35  
 momo 13  
 \*/  
  
 //cout<<"User Input:";* pair<int,int>n;  
 cin>>n.first>>n.second;*//2 3* cout<<n.first<<" "<<n.second<<endl;*//2 3* return 0;  
}**

**Pair Problem:**

1. Codeforce=44A(Indian Summer)
2. CodeForce=166A(Rank List)
3. Geeksforgeeks=Count distinct pairs with difference k

# Almost everything about STL Map

**#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();  
  
 map<string ,int>m;*//map<index,value>* m["hamid"]=1;  
 m["hosen"]=2;  
 m["azad"]=3;  
 m["fahim"]=9;  
  
 cout<<m["fahim"]<<endl<<endl;*//9* map<string,string>genter;  
 genter["hamid"]="male";  
 genter["momo"]="female";  
  
 cout<<genter["hamid"]<<" "<<genter["momo"]<<endl<<endl;*//male female* map<string,int>p1;  
 cout<<p1["hamid"]<<endl<<endl;*//0* map<string,string>p2;  
 cout<<p2["hamid"]<<endl<<endl;*//* vector<long long >v={1,2324352352,452234345,2324352352,2324352352};  
 map<long long ,int>p3;  
  
 *//for(int i=0;i<v.size();i++) p3[v[i]]++;* for(auto u:v)p3[u]++;  
 cout<<p3[2324352352]<<endl<<endl;*//3* vector<long long >v1={1,2324352352,452234345,2324352352,2324352352};  
 map<long long ,int>p4;  
  
 for(auto u:v1) p4[u]++;  
 for(auto u:p4) cout<<u.first<<" "<<u.second<<endl;  
 cout<<endl<<endl;  
 */\*  
 1 1  
 452234345 1  
 2324352352 3  
 \*/* map<string,int>id;  
  
 id["shahriar"]=1;*//log2(n)* id["momo"]=3;  
 id["sharif"]=5;  
 id["prety"]=5;  
 id["prety"]=6;  
 id["dima"]=9;  
  
 for(auto u:id) cout<<u.first<<" "<<u.second<<endl;  
 cout<<endl<<endl;  
 */\*  
 dima 9  
 momo 3  
 prety 6  
 shahriar 1  
 sharif 5  
 \*/* map<int,string>name;  
 name[1]="Hamid";  
 name[2]="Hosen";*//log2(size of map)* name[3]="Azad";  
  
 cout<<name[3]<<endl<<endl;*//log2(3)//Azad  
  
 //unique value* map<int,bool>vis;  
 vector<int>v2={2,2,1,1,3};  
  
 for(auto u:v2) vis[u]=1;*//n\*log2(n)* for(auto u:vis) cout<<u.first<<" "<<u.second<<endl;*//male male* cout<<endl<<endl;  
 */\*  
 1 1  
 2 1  
 3 1  
 \*/* map<string,string> gen;  
 gen["Hamid"]="male";  
 gen["Hosen"]="male";  
 gen["momo"]="female";  
  
 cout<<gen["Hamid"]<<" "<<gen["Hosen"]<<endl<<endl;  
  
 map<string,int>m1;  
 m1.insert(make\_pair("Hamid",21));  
 m1.insert(pair<string,int>("Hosen",22));  
  
 cout<<m1["Hamid"]<<" "<<m1["Hosen"]<<endl<<endl;*//21 22* vector<map<string,int>>v3;  
  
 map<string ,int>c;  
 c["Hamid"]=3;  
 v3.push\_back(c);  
 cout<<v3[0]["Hamid"]<<endl<<endl;*//3* map<int,int>h;  
 h[1]=1;  
 h[2]=2;  
 h[3]=1;  
 h[3]=2;  
  
 cout<<h.size()<<endl<<endl;*//3* return 0;  
}**

**Map Problem:**

1. Geeks for Geeks=Find the Frequency
2. Geeks for Geeks=Twice Counter
3. Geeks for Geeks = Word with maximum frequency
4. AtCoder = C-Good sequence
5. CodeForce=855A(Tom Riddle’s Diary)
6. CodeForce=4C(Registration System)
7. CodeForce=918B(Radio Station)
8. CodeForce=903C(Boxes Packing)
9. SPOJ=RPLD-Database

# Almost everything about STL Set

**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;  
}**

**#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;  
}**

**Set Problem:**

1. Geeks for Geeks=C++ STL | Set 6 (set)
2. Geeks for Geeks=Set Operations
3. HackerRank=Sets-STL
4. LeetCode=217(Contains Duplicate)
5. CodeForce=22A(Second Order Statistics)
6. CodeForce=469A(I Wanna Be the Guy)
7. CodeForce=1325B(CopyCopyCopyCopyCopy)

# STL Multiset

**#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();  
  
 multiset<int>s;  
  
 s.insert(1);  
 s.insert(2);  
 s.insert(1);  
 s.insert(3);  
 s.insert(1);  
 s.insert(4);  
 s.insert(1);  
 s.insert(4);  
  
 cout<<s.size()<<endl;*//8* for(auto u:s){  
 cout<<u<<" ";*//1 1 1 1 2 3 4 4* }  
 cout<<endl<<s.count(4)<<endl;*//2* s.erase(1);  
 for(auto u:s) cout<<u<<" ";*//2 3 4 4* cout<<endl<<endl;  
  
 multiset<int>s1;  
 multiset<int>::iterator it;  
 s1.insert(1);  
 s1.insert(2);  
 s1.insert(1);  
 s1.insert(3);  
 s1.insert(1);  
 s1.insert(4);  
 s1.insert(1);  
 s1.insert(4);  
*/\*  
 auto it=s1.find(1);  
 s1.erase(it);  
 for(auto u:s1) cout<<u<<" ";//1 1 1 2 3 4 4  
 cout<<endl<<endl;  
 \*/* it=s1.find(1);  
 s1.erase(it);  
 for(auto u:s1) cout<<u<<" ";*//1 1 1 2 3 4 4* cout<<endl<<endl;  
 return 0;  
}**

**MultiSet Problem:**

1. Geeks for Geeks=Multiset Operations

# STL Stack

**stack<int>st;  
st.push(1);  
st.push(2);  
st.push(3);  
st.push(4);  
st.push(5);  
  
cout<<"SIZE:"<<st.size()<<endl;  
  
cout<<st.top()<<endl;*//5*st.pop();*//top element delete*cout<<st.top()<<endl;*//4*cout<<"Stack all element print:"<<endl;  
  
while (!st.empty()){  
 cout<<st.top()<<endl;  
 st.pop();  
}*//4 3 2 1***

**MultiSet Problem:**

1. Geeks for Geeks= **Special Stack**
2. Hackerearth=balanced brackets
3. LeetCode=682(Baseball Game)
4. LeetCode=1047(Remove All Adjacent Duplicates In String)

# Almost everything about Queue, Deque

# Queue

**queue<int>q;  
q.push(1);  
q.push(2);  
q.push(3);  
q.push(4);  
q.push(5);  
  
cout<<q.front()<<endl;*//1*q.pop();  
  
cout<<q.front()<<endl<<endl;*//2*cout<<"Queue size:"<<q.size()<<endl<<endl;*//4*cout<<"Queue all element print:"<<endl;  
while (!q.empty()){  
cout<<q.front()<<endl;  
q.pop();  
}  
cout<<endl<<endl;  
*/\*  
 2  
 3  
 4  
 5  
 \*/***

**Queue Problem:**

1. Geeks for Geeks=Queue Push & Pop
2. Geeks for Geeks=Reverse First K elements of Queue
3. Geeks for Geeks=Queue Reversal

# Deque

**deque<int>d;  
d.push\_front(1);  
d.push\_front(2);  
d.push\_back(3);  
d.push\_front(4);  
d.push\_back(5);  
  
cout<<d.front()<<" "<<d.back()<<endl;*//4 5*cout<<"pop front:";  
d.pop\_front();  
for(auto u:d){  
cout<<d.front()<<" ";  
d.pop\_front();  
}*//2 1 3 5*cout<<endl;  
cout<<endl;  
  
d.push\_front(1);  
d.push\_front(2);  
d.push\_back(3);  
d.push\_front(4);  
d.push\_back(5);  
  
cout<<"pop back:";  
d.pop\_back();  
for(auto u:d){  
cout<<d.back()<<" ";  
d.pop\_back();  
}*//3 1 2 4*cout<<endl;  
  
d.push\_front(1);  
d.push\_front(2);  
d.push\_front(3);  
d.push\_back(4);  
d.push\_back(5);  
  
cout<<d.front()<<" "<<d.back()<<endl;*//3 5*d.pop\_front();  
cout<<d.front()<<" "<<d.back()<<endl;*// 2 5*d.pop\_back();  
cout<<d.front()<<" "<<d.back()<<endl;*// 2 4*cout<<d.size()<<endl;*//3*cout<<d.empty()<<endl;*//0*cout<<d.size()<<endl;**

**Deque Problem:**

1. Geeks for Geeks=Deque Implementations

# Almost everything about STL Priority queue

**#include<bits/stdc++.h>  
using namespace std;  
#define hamid() ios\_base::sync\_with\_stdio(0);cin.tie(0);cout.tie(0);  
#define endl '\n';  
int main()  
{  
 hamid();  
  
 priority\_queue<int> p;  
 p.push(1);  
 p.push(2);  
 p.push(3);  
 p.push(4);  
  
 cout<<"Size:"<<p.size()<<endl;*//4* cout<<p.top()<<endl;*//4* p.pop();  
 cout<<p.top()<<endl;*//3* cout<<endl;  
  
 p.push(1);  
 p.push(2);  
 p.push(3);  
 p.push(4);  
  
 cout<<"Element in :";  
 while (!p.empty() ){  
 cout<<p.top()<<" ";  
 p.pop();  
 }*//4 3 3 2 2 1 1* cout<<endl;  
 cout<<"Size:"<<p.size()<<endl;*//0* cout<<endl;  
  
 cout<<"Comparator use: because acending order:"<<endl;  
 priority\_queue<int,vector<int>,greater<int>>q;  
 q.push(5);  
 q.push(1);  
 q.push(2);  
 q.push(4);  
 q.push(2);  
 q.push(3);  
  
 cout<<"Size:"<<q.size()<<endl;  
 while (!q.empty()){  
 cout<<q.top()<<" ";  
 q.pop();  
 }*//1 2 2 3 4 5* cout<<endl;  
 cout<<endl;  
  
 cout<<"Pair comparator use:"<<endl;  
  
 priority\_queue<pair<int,int>,vector<pair<int,int>>,greater<pair<int,int>>> q1;  
 q1.push({1,2});  
 q1.push({2,3});  
 q1.push({2,4});  
 q1.push({4,4});  
 q1.push({3,4});  
  
 while (!q1.empty()){  
 cout<<q1.top().first<<" "<<q1.top().second<<endl;  
 q1.pop();  
 }  
 */\*  
 \* 1 2  
 \* 2 3  
 \* 2 4  
 \* 3 4  
 \* 4 4  
 \*/* cout<<"Size:"<<q1.size()<<endl;*//0* cout<<endl;  
  
 cout<<"Only pair use:"<<endl;  
 priority\_queue<pair<int,int>> q2;  
 q2.push({1,2});  
 q2.push({2,3});  
 q2.push({2,4});  
 q2.push({4,4});  
 q2.push({3,4}) ;  
  
 while (!q2.empty()){  
 cout<<q2.top().first<<" "<<q2.top().second<<endl;  
 q2.pop();  
 }  
 */\*  
 \* 4 4  
 \* 3 4  
 \* 2 4  
 \* 2 3  
 \* 1 2  
 \*/* return 0;  
}**

**#include<bits/stdc++.h>  
using namespace std;  
#define hamid() ios\_base::sync\_with\_stdio(0);cin.tie(0);cout.tie(0);  
#define endl '\n';  
int main()  
{  
 hamid();  
  
 priority\_queue<pair<int,int>> q2;  
 q2.push({1,-2});  
 q2.push({2,-3});  
 q2.push({2,-4});  
 q2.push({4,-4});  
 q2.push({3,-4}) ;  
  
 while (!q2.empty()){  
 cout<<q2.top().first<<" "<<q2.top().second<<endl;  
 q2.pop();  
 }  
 */\*  
 \* 4 -4  
 \* 3 -4  
 \* 2 -3  
 \* 2 -4  
 \* 1 -2  
 \*/* cout<<endl;  
  
 cout<<"contest program :for 1st value grater and 1st value equal then second value small:"<<endl;  
  
 priority\_queue<pair<int,int>> q;  
 q.push({1,-2});  
 q.push({2,-3});  
 q.push({2,-4});  
 q.push({4,-4});  
 q.push({3,-4}) ;  
  
 while (!q.empty()){  
 cout<<q.top().first<<" "<<q.top().second\*-1<<endl;  
 q.pop();  
 }  
 */\*  
 \* 4 4  
 \* 3 4  
 \* 2 3  
 \* 2 4  
 \* 1 2  
 \*/* cout<<endl;  
   
 return 0;  
}**

**Priority queue Problem:**

1. CodeForce=799B(T-shirt buying)
2. SPOJ=REDARR2-Reduce the array
3. Hackerearth=Monk And Champions League
4. Hackerearth=Monk and Multiplication
5. CodeForce=1353D(Constructing the array)

# Template

/\*In the name of Almighty Allah\*/

///\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Bismillahir Rahmanir Rahim \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*///

#include<bits/stdc++.h>

#include <ext/pb\_ds/assoc\_container.hpp>

#include <ext/pb\_ds/tree\_policy.hpp>

using namespace std;

///\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Template Start Here \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*///

///\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* C o n t a i n e r \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*///

typedef long long ll;

typedef vector<int> vi;

typedef vector<string> vs;

typedef vector<ll> vl;

typedef vector<vi> vvi;

typedef vector<vl> vvl;

typedef pair<int,int> pii;

typedef pair<double, double> pdd;

typedef pair<ll, ll> pll;

typedef vector<pii> vii;

typedef vector<pll> vll;

typedef set<int> si;

typedef map<string,int> msi;

typedef double dl;

typedef string  st;

typedef int32\_t i32;//typedef \_\_int32 i32;

typedef int64\_t i64;//typedef \_\_int64 i64;

#define PB push\_back

#define F first

#define S second

#define MP make\_pair

#define endl '\n'

#define all(a) (a).begin(),(a).end()

#define sz(x) (int)x.size()

#define mid(l,r) ((r+l)/2)

#define left(node) (node\*2)

#define right(node) (node\*2+1)

#define mx\_int\_prime 999999937

#define CY cout<<"YES"<<endl

#define CN cout<<"NO"<<endl

#define rn return 0

///\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* C o n s t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*///

const double PI = acos(-1);

const double eps = 1e-9;//10^-9

const int inf = 2000000000;

const ll infLL = 9000000000000000000;

#define MOD 1000000007

#define mem(a,b) memset(a, b, sizeof(a) )

#define gcd(a,b) \_\_gcd(a,b)

#define sqr(a) ((a) \* (a))

#define sw(a,b) swap(a,b)

#define sor(a) sort(a.begin(),a.end())

#define sorr(a) sort(a.begin(),a.end(),greater<int>())

#define uni(s) unique(s.begin(),s.end())-s.begin()

#define mxdex(a) max\_element(v.begin(),v.end())-v.begin()

#define mindex(a) min\_element(v.begin(),v.end())-v.begin()

#define mxele(a) max\_element(v.begin(),v.end())

#define minele(a) min\_element(v.begin(),v.end())

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

#define fraction(a) cout.unsetf(ios::floatfield); cout.precision(a); cout.setf(ios::fixed,ios::floatfield);

#define file() freopen("input.txt","r",stdin);freopen("output.txt","w",stdout);

typedef vector<int>::iterator vit;

typedef set<int>::iterator sit;

int dx[] = {0, 0, +1, -1};

int dy[] = {+1, -1, 0, 0};

//int dx[] = {+1, 0, -1, 0, +1, +1, -1, -1};

//int dy[] = {0, +1, 0, -1, +1, -1, +1, -1};

//Debugger

template < typename F, typename S >

ostream& operator << ( ostream& os, const pair< F, S > & p )

{

    return os << "(" << p.first << ", " << p.second << ")";

}

template < typename T >

ostream &operator << ( ostream & os, const vector< T > &v )

{

    os << "{";

    for(auto it = v.begin(); it != v.end(); ++it)

    {

        if( it != v.begin() ) os << ", ";

        os << \*it;

    }

    return os << "}";

}

template < typename T >

ostream &operator << ( ostream & os, const set< T > &v )

{

    os << "[";

    for(auto it = v.begin(); it != v.end(); ++it)

    {

        if( it != v.begin() ) os << ", ";

        os << \*it;

    }

    return os << "]";

}

template < typename T >

ostream &operator << ( ostream & os, const multiset< T > &v )

{

    os << "[";

    for(auto it = v.begin(); it != v.end(); ++it)

    {

        if( it != v.begin() ) os << ", ";

        os << \*it;

    }

    return os << "]";

}

template < typename F, typename S >

ostream &operator << ( ostream & os, const map< F, S > &v )

{

    os << "[";

    for(auto it = v.begin(); it != v.end(); ++it)

    {

        if( it != v.begin() ) os << ", ";

        os << it -> first << " = " << it -> second ;

    }

    return os << "]";

}

#define dbg(args...) do {cerr << #args << " : "; faltu(args); } while(0)

void faltu ()

{

    cerr << endl;

}

template <typename T>

void faltu( T a[], int n )

{

    for(int i = 0; i < n; ++i) cerr << a[i] << ' ';

    cerr << endl;

}

template <typename T, typename ... hello>

void faltu( T arg, const hello &... rest)

{

    cerr << arg << ' ';

    faltu(rest...);

}

///\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Template End Here \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*///

int main()

{

    optimize();

//freopen("input.txt", "r", stdin);

//freopen("output.txt", "w", stdout);

    rn;

}