**PAIR**

///PAIR --- It is like structure.But it is built in....

///Declare ---------- pair<1st data-type,2nd data-type>variavle name;

pair<int,int>p;

p.first = 1;

p.second = 3;

cout<<p.first<<" "<<p.second<<endl; ///1 and 3

pair<string,int>s;

s.first = "Tusar";

s.second = 39;

cout<<s.first<<" "<<s.second<<endl; ///Tusar and 39

pair<string,vector<int>>t;

t.first = "Tusar";

t.second = {1,2,3};

cout<<t.first<<" ";

for(auto i : t.second){ ///Tusar and 1 2 3

cout<<i<<" ";

}

cout<<endl;

/// make two number pair use make function

pair<int,int>p;

p = make\_pair(1,9);

//or

// p = {1,9}

cout<<p.first<<" "<<p.second<<endl; ///1 and 9

p.first++; /// 1 increment

cout<<p.first<<" "<<p.second<<endl; ///2 and 9

pair<string,vector<int>>p;

p = {"ok" , {8,9}};

cout<<p.first<<" "<<p.second.size()<<endl; /// ok and 2

/// compare two pair

pair<int, int> p1,p2;

p1 = { 2, 3 };

p2 = { 3, 1 };

if(p1<p2) cout<<"YES\n"; /// YES

pair<int, int> p1 = { 2, 3 };

pair<int, int> p2 = { 1, 6 };

/// Getting minimum of 2 pairs

pair<int, int>p = min ( p1, p2 );

cout << p.first << " " << p.second << endl; /// 1 6

/// Getting maximum of 2 pairs

p = max ( p1, p2 );

cout << p.first << " " << p.second << endl; /// 2 3

/// Sorting pair of integers increasing order

vector<pair<int,int>> v;

v.push\_back ( { 1, 5 } );

v.push\_back ( { 2, 5 } );

v.push\_back ( { 7, 1 } );

v.push\_back ( { 3, 6 } );

v.push\_back ( { 3, 6 } );

v.push\_back ( { 7, 0 } );

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

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

cout << endl;

/\*\*

1 5

2 5

3 6

3 6

7 0

7 1

\*/

/// Sorting pair of integers decreasing order

vector<pair<int,int>> v;

v.push\_back ( { 1, 5 } );

v.push\_back ( { 2, 5 } );

v.push\_back ( { 7, 1 } );

v.push\_back ( { 3, 6 } );

v.push\_back ( { 3, 6 } );

v.push\_back ( { 7, 0 } );

sort ( v.rbegin(), v.rend() );

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

cout << endl;

/\*\*

7 1

7 0

3 6

3 6

2 5

1 5

\*/

array

pair<int, int> p[] = {{6,5},{2,3},{4,5},{6,1},{1,9}};

sort(p,p+5);

for(int i=0;i<5;i++)

cout<<p[i].first<<" "<<p[i].second<<endl;

/\*\*1 9

2 3

4 5

6 1

6 5

\*/

///different data type

vector<pair<string,int>> v;

v.push\_back ( { "shahriar", 21 } );

v.push\_back ( { "momo", 13 } );

v.push\_back ( { "sharif", 34 } );

v.push\_back ( { "shahriar", 35 } );

v.push\_back ( { "sharif", 34 } );

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

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

cout << endl;

/\*\*

momo 13

shahriar 21

shahriar 35

sharif 34

sharif 34

\*/

/// Sorting pair of integers

vector<pair<**int**,**int**>> v;

v.push\_back ( { 1, 5 } );

v.push\_back ( { 2, 5 } );

v.push\_back ( { 7, 1 } );

v.push\_back ( { 3, 6 } );

v.push\_back ( { 3, 6 } );

v.push\_back ( { 7, 1 } );

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

**int** Sz = unique ( v.begin(), v.end() ) - v.begin();

cout << Sz << endl;

**for** ( **int** i = 0; i < Sz; i++ ) cout << v[i].first << " " << v[i].second << endl;

cout << endl;

*/\*\**

*4*

*1 5*

*2 5*

*3 6*

*7 1*

*\*/*

vector<pair<string,int>> v;

v.push\_back ( { "shahriar", 21 } );

v.push\_back ( { "momo", 13 } );

v.push\_back ( { "sharif", 34 } );

v.push\_back ( { "shahriar", 35 } );

v.push\_back ( { "sharif", 34 } );

v.push\_back ( { "shahriar", 21 } );

v.push\_back ( { "momo", 13 } );

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

int Sz = unique ( v.begin(), v.end() ) - v.begin();

for ( int i = 0; i < Sz; i++ ) cout << v[i].first << " " << v[i].second << endl;

/\*\*

momo 13

shahriar 21

shahriar 35

sharif 34

\*/

///user input

pair<int,int> p;

cin>>p.first>>p.second;

cout<<p.first<<" "<<p.second<<endl;

/\*\*

2 3

2 3

\*/

Important

Nijer iccha moto condition set kore sort kora

bool cmp ( const pair<int,int> &p1, const pair<int, int> &p2 )

{

if ( p1.first > p2.first ) return 1;

if ( p1.first == p2.first ) return ( p1.second < p2.second );

return 0;

}

int main(){

tusar();

vector<pair<int,int>> v;

v = {{4,143},{3,120},{4,500},{1,20}};

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

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

cout << endl;

/\*\*

4 143

4 500

3 120

1 20

\*/

Without function

v = { {2, 3}, {4, 5}, {1, 5}, {1, 6}, {6, 7}, {6, 8} };

**for** ( **int** i = 0; i < v.size(); i++ ) v[i].first \*= -1;

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

**for** ( **auto** u : v ) cout << (u.first\*-1) << " " << u.second << endl;

cout << endl;

*/\*\**

*6 7*

*6 8*

*4 5*

*2 3*

*1 5*

*1 6*

*\*/*