**CSE2012**

**DAA LAB**

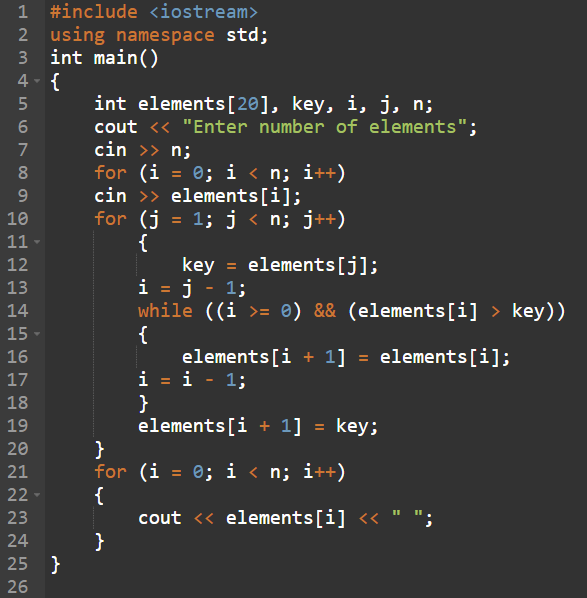
Name: Preyash

Registration Number: 20BPS1022

Ex1: Insertion Sort

**Task 1: Array Method**

**Code window:**



**Code:**

#include <iostream>

using namespace std;

int main()

{

int elements[20], key, i, j, n;

cout << "Enter number of elements";

cin >> n;

for (i = 0; i < n; i++)

cin >> elements[i];

for (j = 1; j < n; j++)

{

key = elements[j];

i = j - 1;

while ((i >= 0) && (elements[i] > key))

{

elements[i + 1] = elements[i];

i = i - 1;

}

elements[i + 1] = key;

}

for (i = 0; i < n; i++)

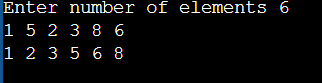
{

cout << elements[i] << " ";

}

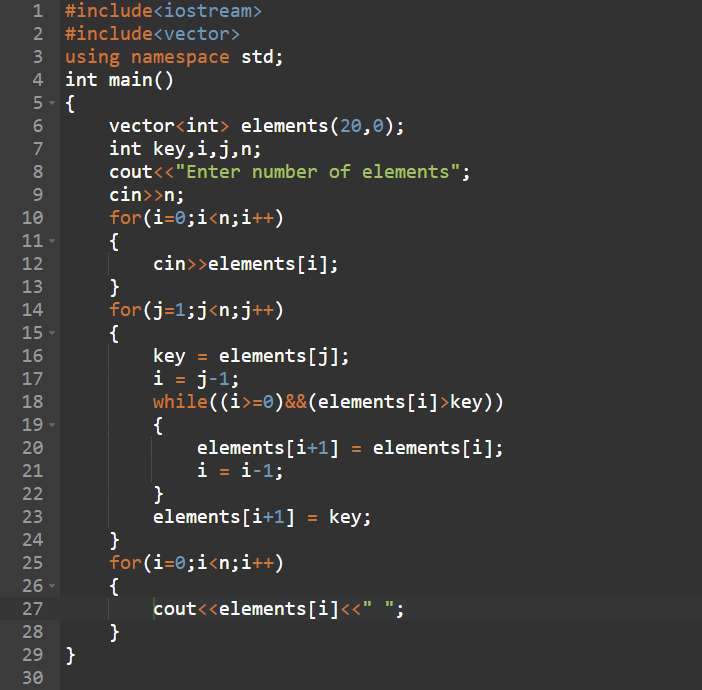
}

**Output:**



**2. Insertion Sort using Vector**

**Code window:**

****

**Code:**

#include<iostream>

#include<vector>

using namespace std;

int main()

{

vector<int> elements(20,0);

int key,i,j,n;

cout<<"Enter number of elements";

cin>>n;

for(i=0;i<n;i++)

{

cin>>elements[i];

}

for(j=1;j<n;j++)

{

key = elements[j];

i = j-1;

while((i>=0)&&(elements[i]>key))

{

elements[i+1] = elements[i];

i = i-1;

}

elements[i+1] = key;

}

for(i=0;i<n;i++)

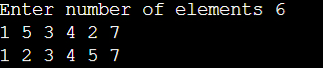
{

cout<<elements[i]<<" ";

}

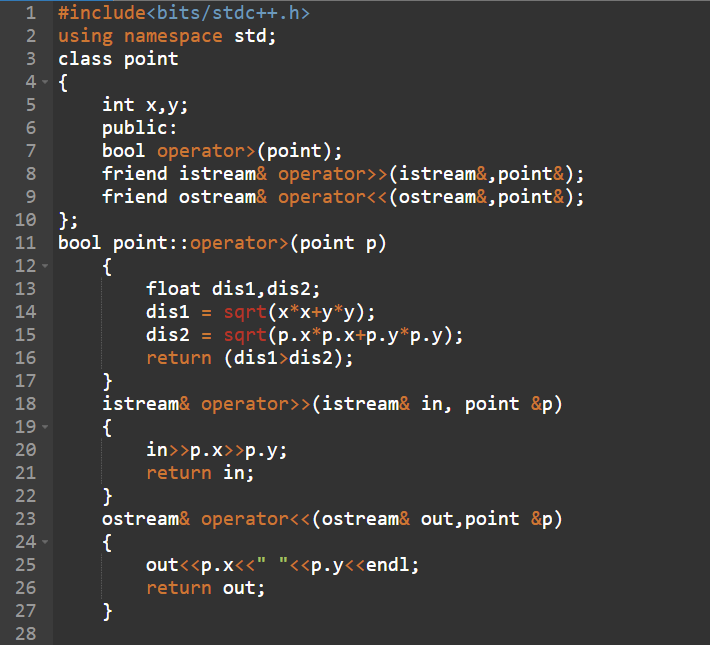
}

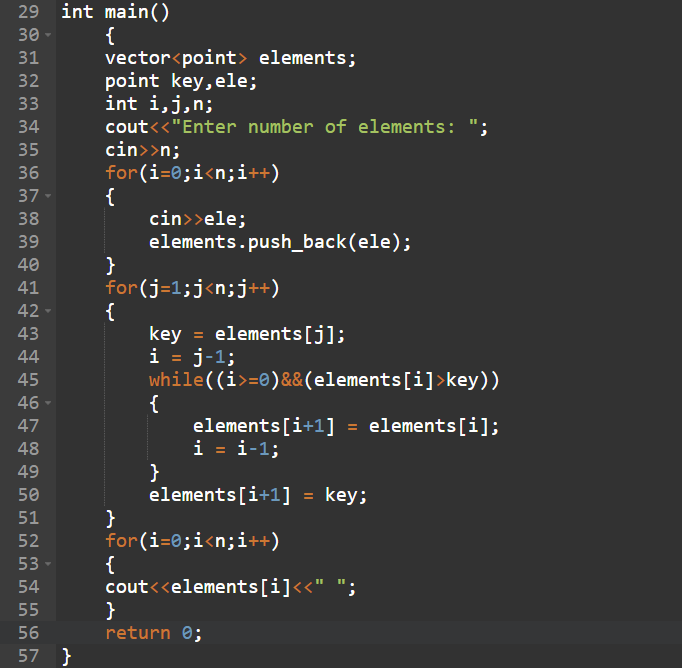
**Output:**

****

**3. Insertion Sort using Point Class**

**Code window:**

****

****

**Code:**

#include<bits/stdc++.h>

using namespace std;

class point

{

int x,y;

public:

bool operator>(point);

friend istream& operator>>(istream&,point&);

friend ostream& operator<<(ostream&,point&);

};

bool point::operator>(point p)

{

float dis1,dis2;

dis1 = sqrt(x\*x+y\*y);

dis2 = sqrt(p.x\*p.x+p.y\*p.y);

return (dis1>dis2);

}

istream& operator>>(istream& in, point &p)

{

in>>p.x>>p.y;

return in;

}

ostream& operator<<(ostream& out,point &p)

{

out<<p.x<<" "<<p.y<<endl;

return out;

}

int main()

{

vector<point> elements;

point key,ele;

int i,j,n;

cout<<"Enter number of elements: ";

cin>>n;

for(i=0;i<n;i++)

{

cin>>ele;

elements.push\_back(ele);

}

for(j=1;j<n;j++)

{

key = elements[j];

i = j-1;

while((i>=0)&&(elements[i]>key))

{

elements[i+1] = elements[i];

i = i-1;

}

elements[i+1] = key;

}

for(i=0;i<n;i++)

{

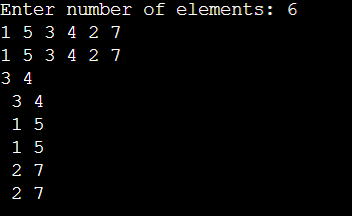
cout<<elements[i]<<" ";

}

return 0;

}

**Output:**

****