**Experiment 4.1**

**4.1)** **Illustrate the use of function template.**

**Aim:** To Illustrate the use of function template.

**Program:**

#include<iostream>

using namespace std;

template <typename T>

void sort(T a[],int n){

T temp;

int i,j;

for(i=0;i<n-1;i++){

for(j=0;j<n-1-i;j++){

if(a[j]>a[j+1]){

temp=a[j];

a[j]=a[j+1];

a[j+1]=temp;}}}

cout<<"sorted array:";

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

cout<<a[i]<<" ";}

cout<<endl;};

int main(){

int a[5]={12,97,34,56,3};

char c[5]={'s','e','a','m','h'};

float f[5]={2.5,14.7,98.2,33.2,85.3};

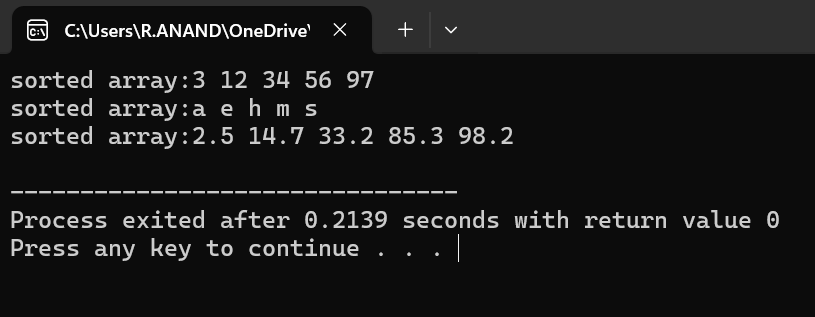
sort (a,5);

sort (c,5);

sort (f,5);

return 0;}

**Output:**

****

**Experiment 4.2)** **. Implement template class**

**Aim:** To . Implement the template class

**Program:**

#include<iostream>

using namespace std;

template <class T>

class sample {

private:

T x;

public:

void get() {

cout << "Enter a value: ";

cin >> x; }

void show(){

cout << "x = " << x << endl; }};

int main() {

sample<char> s1;

s1.get();

s1.show();

sample<int> s2;

s2.get();

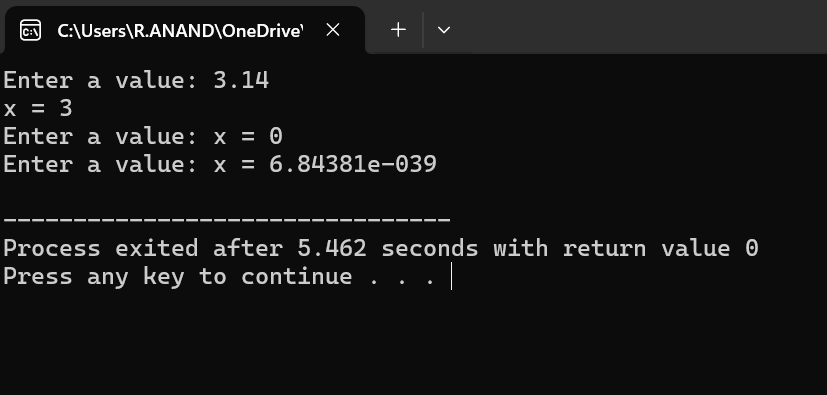
s2.show();

sample<float> s3;

s3.get();

s3.show();

return 0;}

**Output:** ****

**Experiment 4.3**

**4.3)** **Implement class templates with multiple parameters.**

**Aim:** To Implement class templates with multiple parameters.

**Program:**

#include<iostream>

using namespace std;

template <class T1, class T2>

class sample {

T1 a;

T2 b;

public:

void getdata() {

cout << "Enter a and b: " << endl;

cin >> a >> b; }

void display() {

cout << "Displaying the values: " << endl;

cout << "a = " << a << endl;

cout << "b = " << b << endl; }};

int main() {

sample<int, int> s1;

sample<int, char> s2;

sample<int, float> s3;

cout << "Two integer data: " << endl;

s1.getdata();

s1.display();

cout << "Integer and character data: " << endl;

s2.getdata();

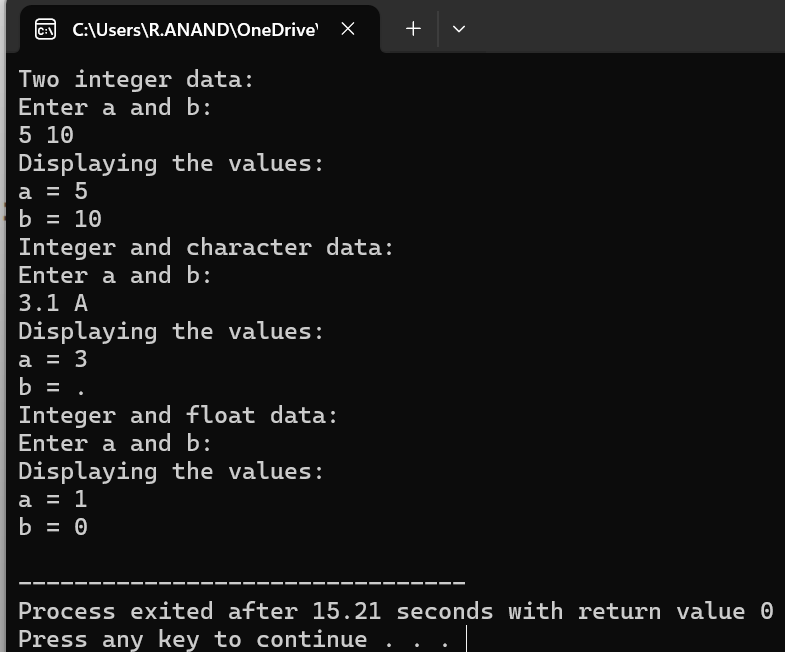
s2.display();

cout << "Integer and float data: " << endl;

s3.getdata();

s3.display();

return 0;}

**Output:** ****