

Theory Questions

1) Throwaway prototyping

Incremental prototyping

Extreme Prototyping

Evolutionary Prototyping

2) There are four parameter passing mechanisms in C++,

Call by value :- The values are copied into newly created variables.

Call by reference :- The original variables are referred using another name.

Call by Address :- The addresses of the variables are sent to the called function.

Return by Reference :- The reference to a variable is sent back and hence function call on left hand side is possible.

3) The reference variables are nothing but new names to the existing variables, hence data is not used up for reference variables. In its place we can use pointer also but pointer occupies memory. Hence using reference variables we can both edit the values in the called function and save the memory also.

4) As mentioned in the above answer, using a reference variable we can save the memory occupied by the pointer and also we have no confusion over dereferencing it. A pointer has to be dereferenced but not a reference variable.

5) Function calling usually takes a lot of time. When a function is called the control has to be transferred to the function and if it returns a value, it will take further more time. So as to save this time, the function is made inline, wherever the function is called, that part of code is replaced by the function definition.

6) Wherever an inline function is called, that part of code is replaced by the function definition. Macro need not be the same. An inline function may have a return type. An inline function can have multiple statements in its definition.

7) We use default arguments in functions so as to send the parameters to the called function. Usually it generates an error in C but in C++ we have an option to execute the called function without sending the parameters, for example the value of pi can be sent by default. We send the default arguments whenever some values are taken by default (like the value of pi).

8) The existence of two or more functions of the same name is known as function overloading. The function when called gets executed based upon the number of arguments passed, type of arguments passed and the return type of the function.

9) The Generic Programming process focuses on finding commonality among similar implementations of the same algorithm, then providing suitable abstractions so that a single, generic algorithm can cover many concrete implementations. In C++, class and function templates are particularly effective mechanisms for generic programming because they make the generalization possible without sacrificing efficiency.

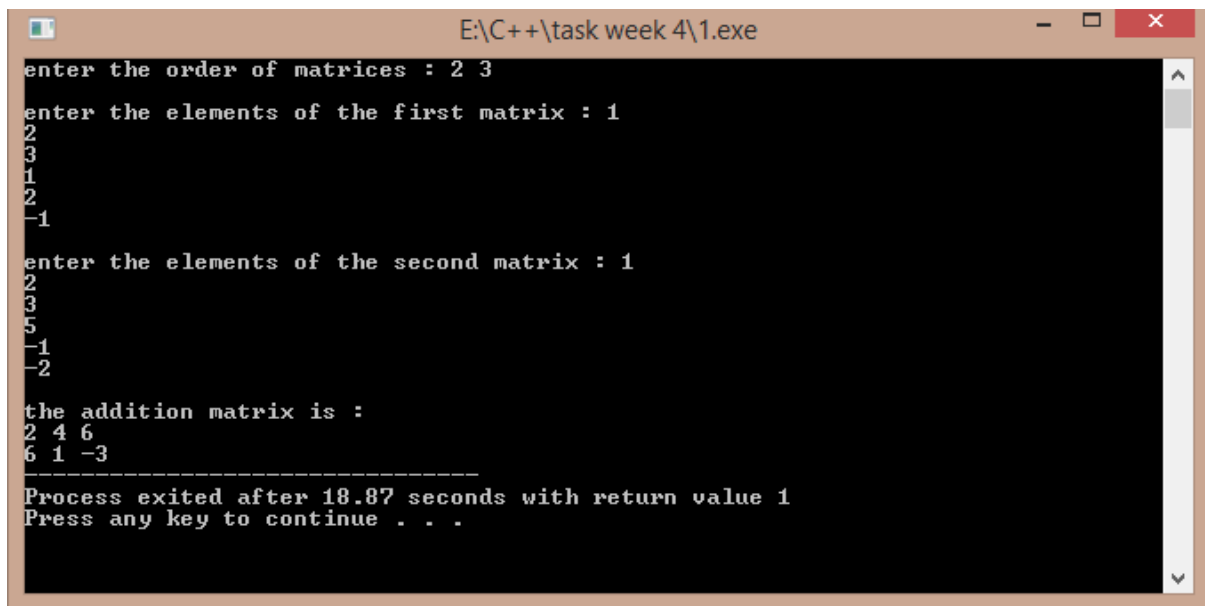
10) There is no way for the compiler to verify that the macro parameters are of compatible types. The macro is expanded without any special type checking. Templates are easier to write. You create only one generic version of your class or function instead of manually creating specializations. Template allows the functions and classes to operate with generic types. This will allow a function to access various methods or operations without rewriting them. Macro parameters cannot be verified.

11) Templates provide an advantage when you want to perform the same action on types that can be different. You can use overloading when you want to apply different operations depending on the type. This is a major difference between them.

Programming Questions

```
1) #include<iostream>
using namespace std;
int main()
{
    cout<<"enter the order of matrices : ";
    int m,n,i,j;
    cin>>m>>n;
    int p[10][10],q[10][10],r[10][10];
    if(m>10||n>10) //an exception if the array size is huge
    {
        cout<<"size overload...";
        return -1;
    }
    cout<<"\nenter the elements of the first matrix : ";
    for(i=0;i<m;i++)
    for(j=0;j<n;j++)
    cin>>p[i][j];
    cout<<"\nenter the elements of the second matrix : ";
    for(i=0;i<m;i++) //input of matrices
    for(j=0;j<n;j++)
    cin>>q[i][j];
    for(i=0;i<m;i++)
    for(j=0;j<n;j++) //addition
    r[i][j]=p[i][j]+q[i][j];
    cout<<"\nthe addition matrix is : ";
    for(i=0;i<m;i++)
    {
        cout<<endl;
        for(j=0;j<n;j++)
        cout<<r[i][j]<<" "; //display of the result matrix
    }
    return 1;
```

}



```
E:\C++\task week 4\1.exe
enter the order of matrices : 2 3
enter the elements of the first matrix : 1
2
3
1
2
-1
enter the elements of the second matrix : 1
2
3
5
-1
-2
the addition matrix is :
2 4 6
6 1 -3
-----
Process exited after 18.87 seconds with return value 1
Press any key to continue . . .
```

2)

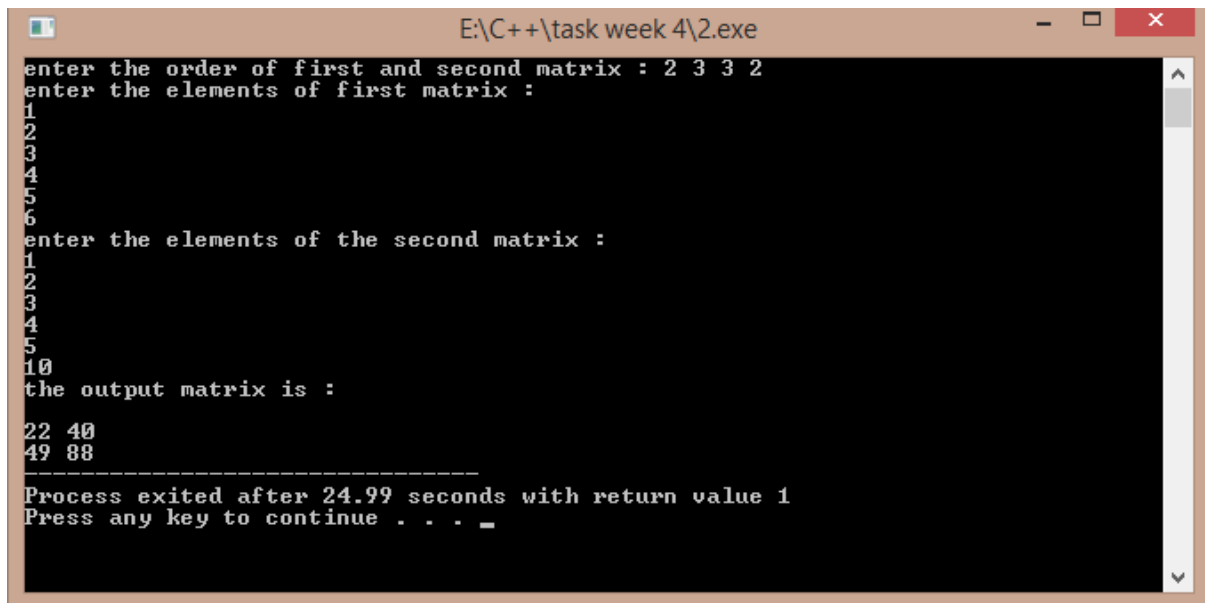
```
#include<iostream>
using namespace std;
void multi(int p[10][10],int q[10][10],int r[10][10],int m,int n,int q1)
{
    //a function to find out the product of matrices
    int i,j,k;
    for(i=0;i<m;i++)
    {
        for(j=0;j<q1;j++)
        {
            r[i][j]=0;
            for(k=0;k<n;k++)
            r[i][j]+=p[i][k]*q[k][j];
        }
    }
}
int main()
{
```

```

int p[10][10],q[10][10],r[10][10];int m,n,p1,q1,i,j;
cout<<"enter the order of first and second matrix : ";
cin>>m>>n>>p1>>q1;
if(p1>10||q1>10||m>10||n>10)
{
    cout<<"size exceeded ...";
    return -1;
}
if(n!=p1)
{
    //an exception if there is a wrong input of order
    cout<<"wrong order input ...";
    return -1;
}
cout<<"enter the elements of first matrix : \n";
for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)                //input of matrix
        cin>>p[i][j];
}
cout<<"enter the elements of the second matrix : \n";
for(i=0;i<p1;i++)
{
    for(j=0;j<q1;j++)
        cin>>q[i][j];
}
multi(p,q,r,m,n,q1);
cout<<"the output matrix is : \n";
for(i=0;i<m;i++)
{
    cout<<endl;                //output of the multiplication matrix
    for(j=0;j<q1;j++)
        cout<<r[i][j]<<" ";
}

```

```
    return 1;
}
```



```
E:\C++\task week 4\2.exe
enter the order of first and second matrix : 2 3 3 2
enter the elements of first matrix :
1
2
3
4
5
6
enter the elements of the second matrix :
1
2
3
4
5
10
the output matrix is :
22 40
49 88
-----
Process exited after 24.99 seconds with return value 1
Press any key to continue . . . _
```

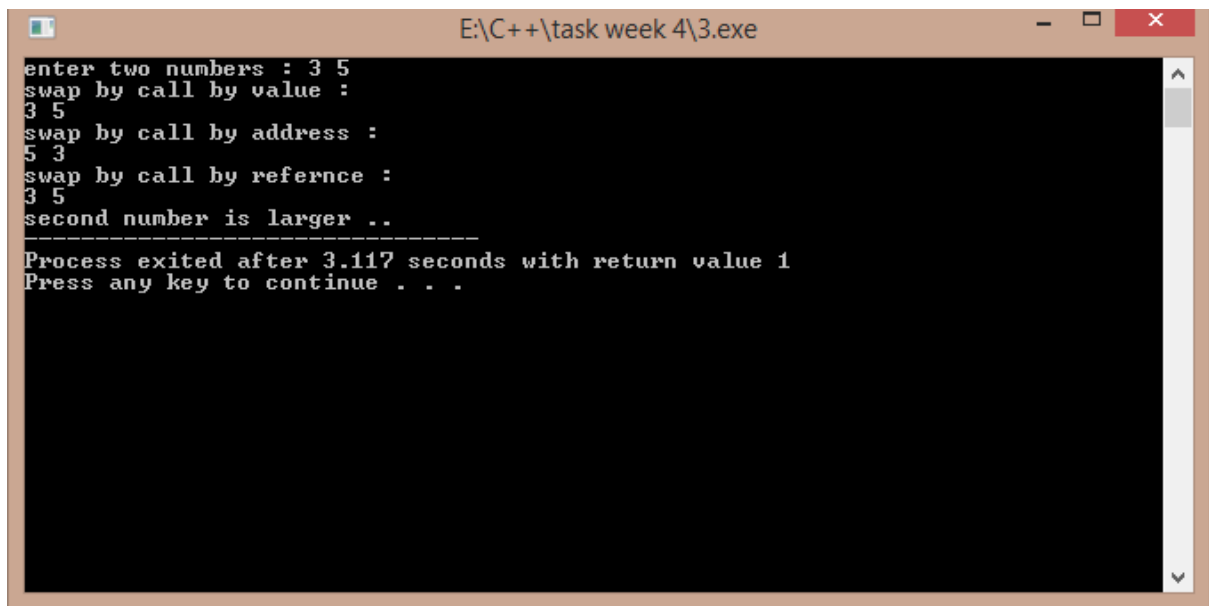
3)

```
#include<iostream>
using namespace std;
void call_by_value(int a,int b)
{
    int t;
    t=a;
    a=b;
    b=t;
}
void call_by_address(int *a,int *b)
{
    int t;
    t=*a;
    *a=*b;
    *b=t;
}
void call_by_reference(int &x,int &y)
{
    int t;
```

```

        t=x;
        x=y;
        y=t;
    }
    int &return_by_reference(int a,int b)
    {
        if(a>b)
            return a;
        else
            return b;
    }
    int main()
    {
        cout<<"enter two numbers : ";
        int a,b,c;
        cin>>a>>b;
        cout<<"swap by call by value : \n"; //function calls to represent each mechanism
        call_by_value(a,b);
        cout<<a<<" "<<b<<endl;
        cout<<"swap by call by address : \n";
        call_by_address(&a,&b);
        cout<<a<<" "<<b<<endl;
        cout<<"swap by call by refernce : \n";
        call_by_reference(a,b);
        cout<<a<<" "<<b<<endl;
        return_by_reference(a,b)=-4; //the larger value is assigned the value of -4
        if(a==-4) //by return by reference mechanism
            cout<<"first number is larger ..";
        else
            cout<<"second number is larger ..";
        return 1;
    }

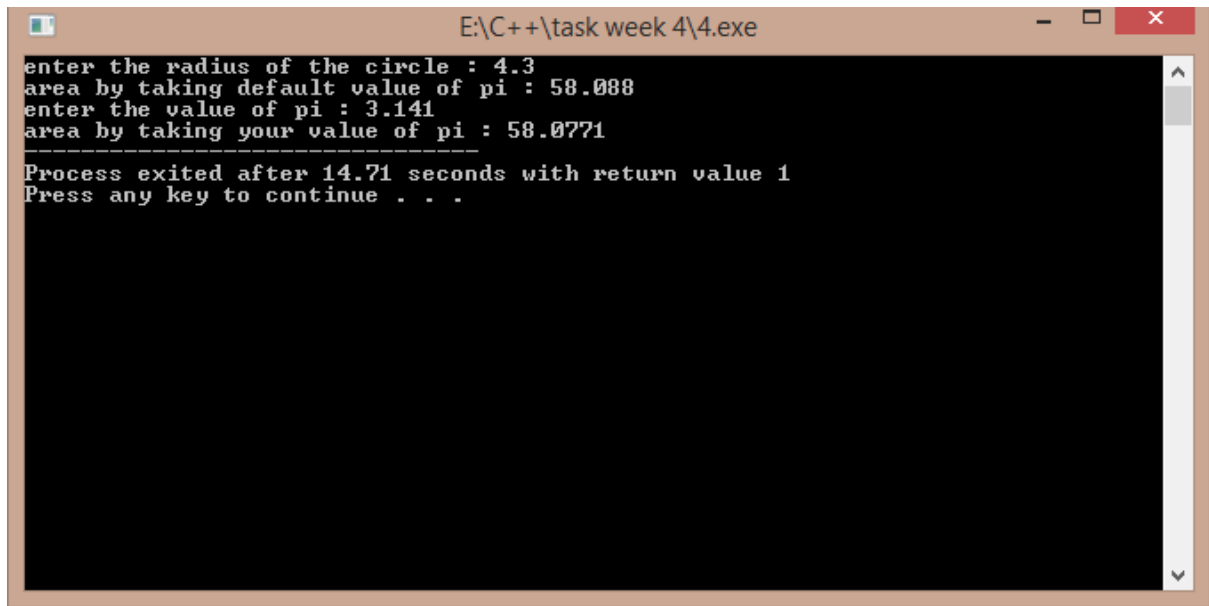
```



```
enter two numbers : 3 5
swap by call by value :
3 5
swap by call by address :
5 3
swap by call by refernce :
3 5
second number is larger ..
-----
Process exited after 3.117 seconds with return value 1
Press any key to continue . . .
```

4)

```
#include<iostream>
using namespace std;
int main()
{
    void area(float r,float pi=3.141592); //default value of pi is taken
    float a,pi;                          //if the pi value is not passed
    cout<<"enter the radius of the circle : ";
    cin>>a;
    cout<<"area by taking default value of pi : ";
    area(a);
    cout<<"\nenter the value of pi : ";
    cin>>pi;
    cout<<"area by taking your value of pi : ";
    area(a,pi);
    return 1;
}
void area(float r,float pi)
{
    cout<<(pi*r*r);
}
```

```
E:\C++\task week 4\4.exe
enter the radius of the circle : 4.3
area by taking default value of pi : 58.088
enter the value of pi : 3.141
area by taking your value of pi : 58.0771
-----
Process exited after 14.71 seconds with return value 1
Press any key to continue . . .
```

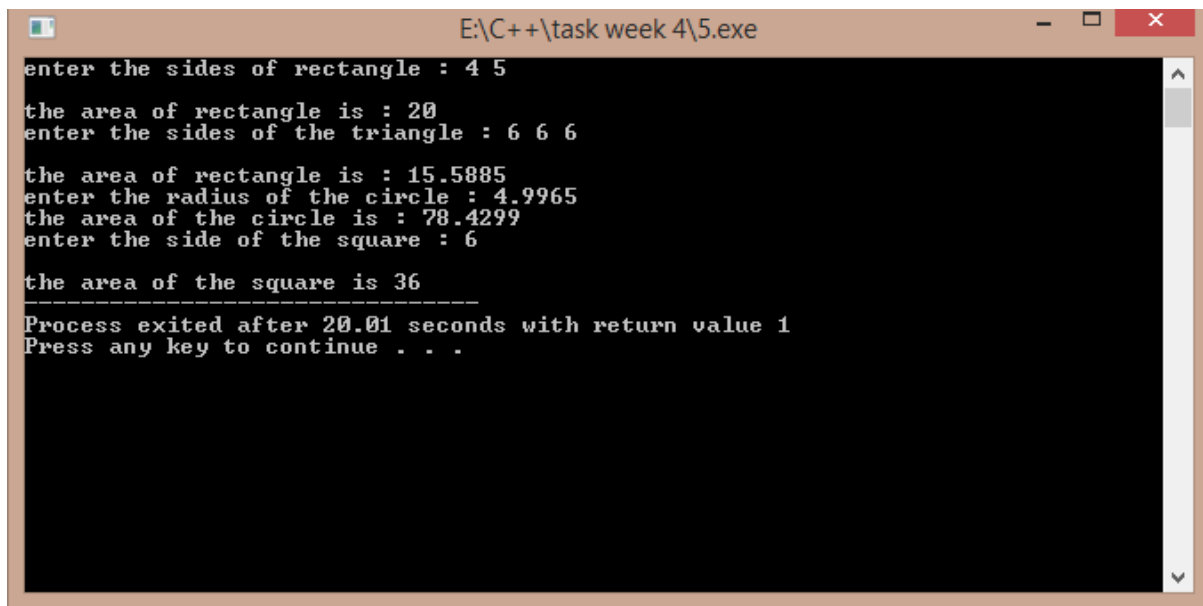
5)

```
#include<iostream>
#include<cmath>
using namespace std;
float area(float r)
{
    return ((3.141592)*r*r);
}
int area(int s)
{
    return s*s;
}
int area(int a,int b)
{
    return a*b;
}
float area(float a,float b,float c)
{
    float s,ar;
    s=(a+b+c)/2;
    ar=sqrt(s*(s-a)*(s-b)*(s-c));
```

```

        return ar;
    }
    int main()
    {
        int a,b,ar;float r,a1,b1,c1,ar1;
        cout<<"enter the sides of rectangle : ";
        cin>>a>>b;
        ar=area(a,b);
        cout<<"\nthe area of rectangle is : "<<ar<<endl; //calling different functions
        cout<<"enter the sides of the triangle : ";          //with the same name
        cin>>a1>>b1>>c1;
        ar1=area(a1,b1,c1);
        cout<<"\nthe area of rectangle is : "<<ar1<<endl;
        cout<<"enter the radius of the circle : ";
        cin>>r;
        ar1=area(r);
        cout<<"the area of the circle is : "<<ar1<<endl;
        cout<<"enter the side of the square : ";
        cin>>a;
        ar=area(a);
        cout<<"\nthe area of the square is "<<ar;
        return 1;
    }

```



```
E:\C++\task week 4\5.exe
enter the sides of rectangle : 4 5
the area of rectangle is : 20
enter the sides of the triangle : 6 6 6
the area of rectangle is : 15.5885
enter the radius of the circle : 4.9965
the area of the circle is : 78.4299
enter the side of the square : 6
the area of the square is 36
-----
Process exited after 20.01 seconds with return value 1
Press any key to continue . . .
```

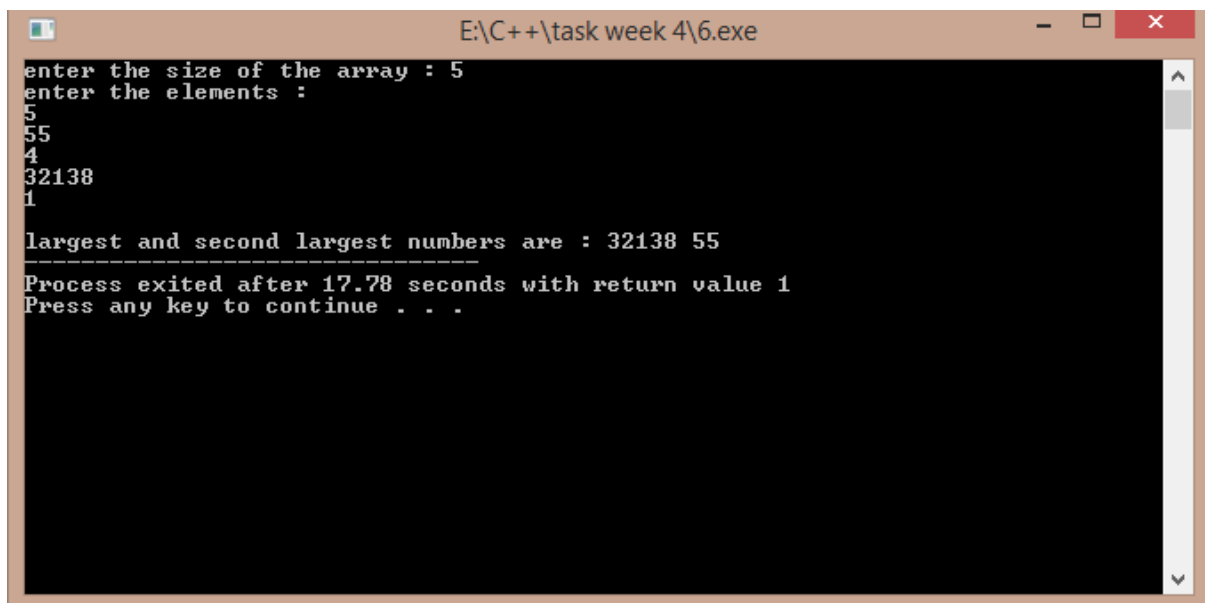
6)

```
#include<iostream>
using namespace std;
template<class m>                //template named m is used
void firsttwo(m *p,m &n) //a template pointer and the same data type reference is used
{
    int i,m1=-1,m2=-1;
    for(i=0;i<n;i++)
    {
        if(m1<p[i])
        {
            m1=p[i];
        }
    }
    for(i=0;i<n;i++)
    {
        if(m2<p[i]&& p[i]!=m1)    //if the found number is not equal to
        {                       //the largest number .
            m2=p[i];
        }
    }
}
```

```

        cout<<"\nlargest and second largest numbers are : "<<m1<<" "<<m2;
    }
int main()
{
    int n,i;
    cout<<"enter the size of the array : ";
    cin>>n;
    int *p=new int[n];
    cout<<"enter the elements : \n";
    for(i=0;i<n;i++)
    {
        cin>>p[i];                //calling the template function
    }
    firsttwo(p,n);
    return 1;
}

```



The screenshot shows a Windows command prompt window titled "E:\C++\task week 4\6.exe". The program prompts the user to "enter the size of the array : 5" and "enter the elements :". The user enters the numbers 5, 55, 4, 32138, and 1. The program then outputs "largest and second largest numbers are : 32138 55". Below this, it shows "Process exited after 17.78 seconds with return value 1" and "Press any key to continue . . .".

7)

```

#include<iostream>
using namespace std;
template<class K>    //template named K is used
void lsearch(K *p,K &x,K n)    //searching array , search element , no of elements

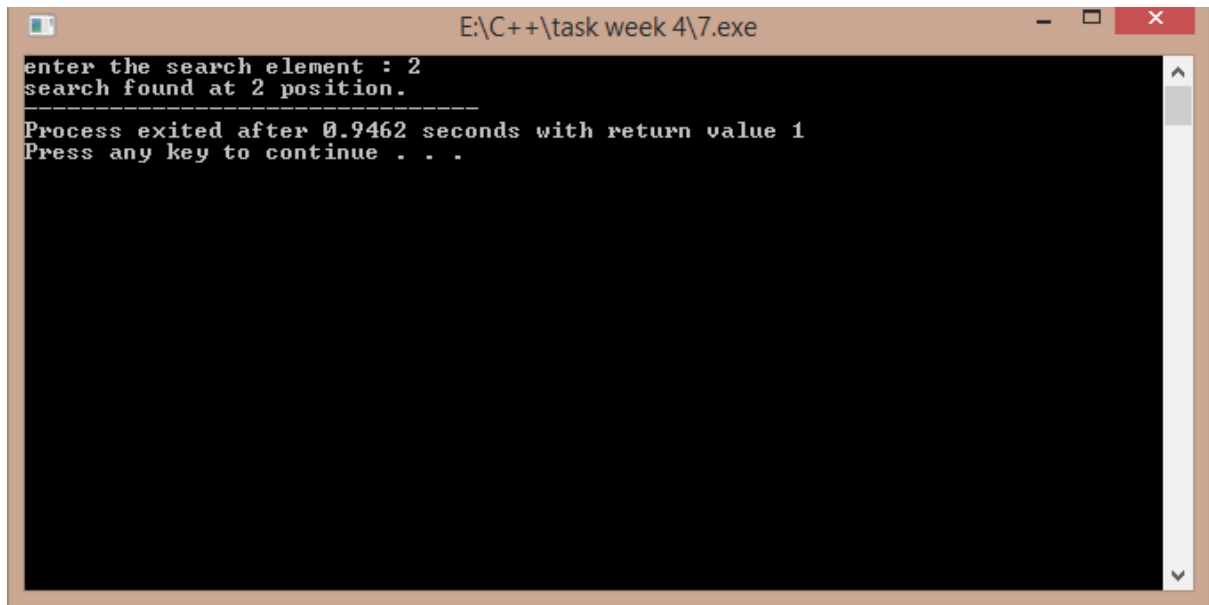
```

```

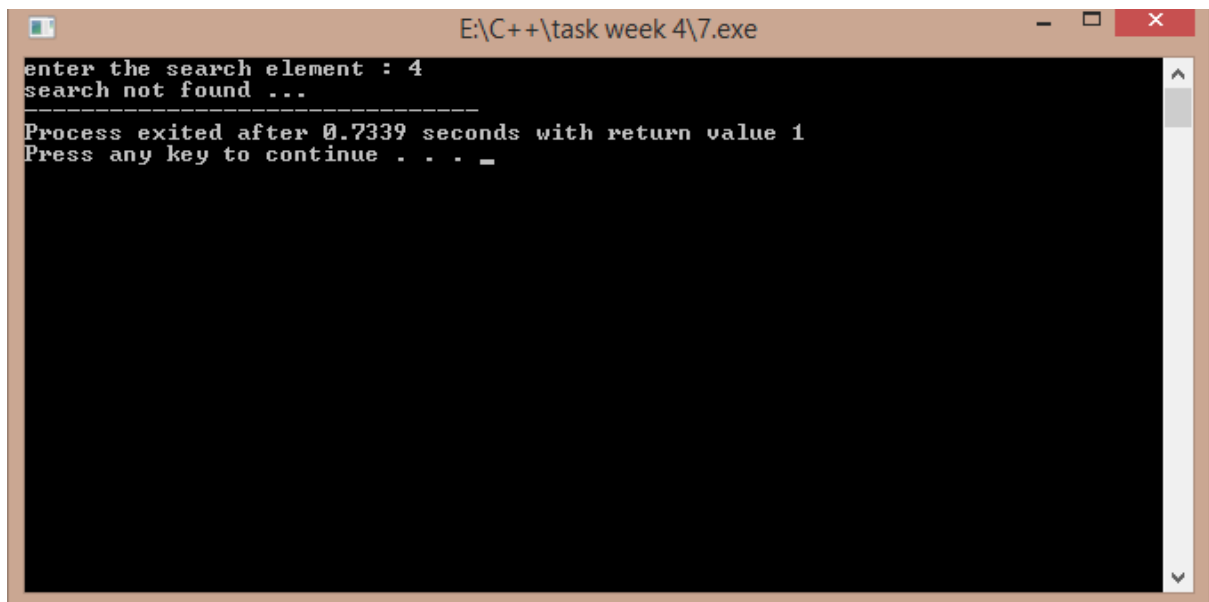
{
    int i,flag=0;
    for(i=0;i<n;i++)
    {
        if(p[i]==x)
        {
            cout<<"search found at "<<i+1<<" position.";
            flag=1;
            break;
        }
    }
    if(flag==0)          //flag variable to check if search is successful or not
        cout<<"search not found ...";
}

int main()
{
    int i,x;
    cout<<"enter the search element : ";
    cin>>x;
    int p[]={1,2,33,44,55,66,77,88,99,110};
    lsearch(p,x,10);
    return 1;
}

```



```
E:\C++\task week 4\7.exe
enter the search element : 2
search found at 2 position.
-----
Process exited after 0.9462 seconds with return value 1
Press any key to continue . . .
```



```
E:\C++\task week 4\7.exe
enter the search element : 4
search not found ...
-----
Process exited after 0.7339 seconds with return value 1
Press any key to continue . . . _
```

8)

```
#include<iostream>
using namespace std;
int n;                                //number of elements declared as global
template<class t>
void sort1(t *a)  //a template function which may receive an array to be sorted of
{
    //of unknown data type.
    int i,j;t k;
    for(i=0;i<n;i++)
    {
```

```

        for(j=0;j<n-i-1;j++)
        {
            if(a[j]>a[j+1])
            {
                k=a[j];
                a[j]=a[j+1];
                a[j+1]=k;
            }
        }
    }
}

int main()
{
    cout<<"enter the data type : 1 for float , 2 for char , 3 for int : ";
    int ch,i;
    cin>>ch;
    cout<<"enter the number of elements : ";
    cin>>n;
    switch(ch)                //switch case used for multiple sorting entry
    {
        case 1:{
            float *p=new float[n];
            cout<<"enter the data : \n";
            for(i=0;i<n;i++)
            cin>>p[i];
            sort1(p);
            cout<<"sorted data is : \n";
            for(i=0;i<n;i++)
            {
                cout<<p[i]<<" ";
            }
            }break;

        case 2:
            {

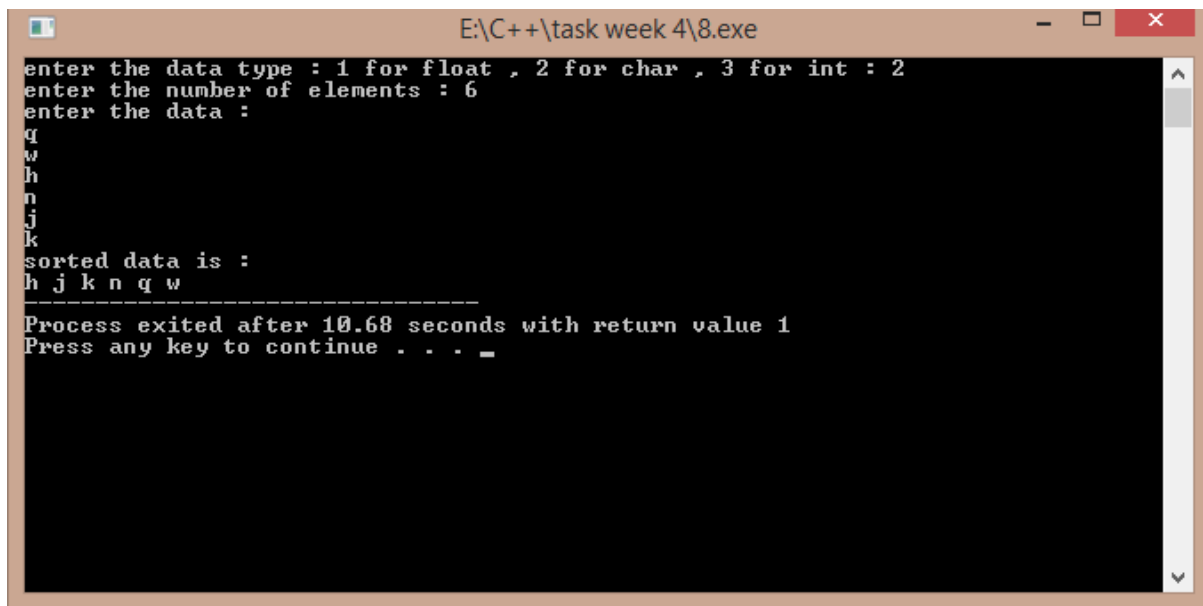
```

```

        char *p=new char[n];
        cout<<"enter the data : \n";
        for(i=0;i<n;i++)
            cin>>p[i];
        sort1(p);
        cout<<"sorted data is : \n";
        for(i=0;i<n;i++)
        {
            cout<<p[i]<<" ";
        }
    }break;
case 3:
    {
        int *p=new int[n];
        cout<<"enter the data : \n";
        for(i=0;i<n;i++)
            cin>>p[i];
        sort1(p);
        cout<<"sorted data is : \n";
        for(i=0;i<n;i++)
        {
            cout<<p[i]<<" ";
        }
    }break;
default : cout<<"wrong option ....";
}

return 1;
}

```

```
enter the data type : 1 for float , 2 for char , 3 for int : 2
enter the number of elements : 6
enter the data :
q
w
h
n
j
k
sorted data is :
h j k n q w
-----
Process exited after 10.68 seconds with return value 1
Press any key to continue . . . _
```

9)

```
#include<iostream>
using namespace std;
void towers(int,char,char,char);
int main()
{
    int n; //Declare the variables to be used
           //Get the input for number of disks
    cout<<"enter the no of disks : ";
    cin>>n;
    towers(n,'A','C','B'); //Call the function
    return 1;
}

void towers(int n,char from,char to,char aux)
{
    if(n==1) // If there is only one disk
    {
        cout<<endl<<"move 1 from peg "<<from<<" to "<<to;
        return;
    }
}
```

```

towers(n-1,from,aux,to); //Recursive Call
cout<<endl<<"move "<<n<<" from peg "<<from<<" to "<<to;
towers(n-1,aux,to,from);
}

```

```

E:\C++\task week 4\9.exe
enter the no of disks : 4
move 1 from peg A to B
move 2 from peg A to C
move 1 from peg B to C
move 3 from peg A to B
move 1 from peg C to A
move 2 from peg C to B
move 1 from peg A to B
move 4 from peg A to C
move 1 from peg B to C
move 2 from peg B to A
move 1 from peg C to A
move 3 from peg B to C
move 1 from peg A to B
move 2 from peg A to C
move 1 from peg B to C
-----
Process exited after 1.135 seconds with return value 1
Press any key to continue . . . _

```

10)

```

#include<iostream>
#include<string.h>
using namespace std;
void sort1(char a[100][100],int n)
{
    int i,j;char temp[100];           //temporary variable to swap
    for(i=0;i<n;i++)
    {
        for(j=0;j<n-i-1;j++)           //using bubble sort concept
        {
            if(strcmp(a[j],a[j+1])>0)
            {
                strcpy(temp,a[j]);
                strcpy(a[j],a[j+1]);
                strcpy(a[j+1],temp);
            }
        }
    }
}

```

```

        }
    }
}

int main()
{
    char a[100][100];int n,i;
    cout<<"enter the number of words : ";
    cin>>n;
    if(n>100)
    {
        cout<<"too many words ...";
        return -1;
    }
    cout<<endl<<"enter "<<n<<" words : ";
    for(i=0;i<n;i++)
    {
        cin>>a[i];                //displaying the words in sorted order
    }
    sort1(a,n);
    cout<<"\nwords in sorted order are : \n";
    for(i=0;i<n;i++)
    cout<<endl<<a[i];
    return 1;
}

```

//output is below

```
E:\C++\task week 4\10.exe
enter the number of words : 6
enter 6 words : sekhar
karedla
anantha
sashi
narain
cbit

words in sorted order are :
anantha
cbit
karedla
narain
sashi
sekhar
=====
Process exited after 19.75 seconds with return value 1
Press any key to continue . . .
```