**Experiment No. 1**

Write a menu-based program that performs following operations:

1. Find the factorial of a number
2. Print the Fibonacci series up to 'n' number
3. Swap elements
4. Find the greatest element in an array
5. Find the first and second greatest element in an array
6. Exit the program

**SOURCE CODE**

#include <iostream>

#include<stdlib.h>

#include<stdio.h>

using namespace std;

void cls()

{ system("cls"); }

int factorial( int n)

{ if(n<=0)

return 1;

else

return((n\*factorial(n-1)));

}

int fibbo(int n)

{ int a=0,b=1,i=0,c;

cout<<"FIBONACCI SERIES ::"<<a<<" "<<b<<" ";

while(i<n-2)

{ c=a+b;

cout<<c<<" ";

a=b;

b=c;

i++;

}

}

int swapi(int &a ,int &b)

{ a=a+b;

b=a-b;

a=a-b;

}

void maximum(int a[],int n)

{ int m=a[0];

for(int i=1;i<n;i++)

{ if(m<a[i])

m=a[i];

}cout<<"MAXIMUM ELEMENT IS:"<<m; }

void second\_maximum(int a[],int n)

{ int m=a[0],m2;

for(int i=1;i<n;i++)

{ if(m<a[i])

{ m2=m;

m=a[i]; }

}cout<<"SECOND MAXIMUM ELEMENT IS:"<<m2;

}

int main()

{ int i,a,b,n,f,fi,n1,n2,r=0,x[100],y[100];

char c='y',g;

while(c=='y'||c=='Y')

{ if(r!=0)

cls();

cout<<"PRESS 1 FACTORIAL OF A GIVEN NUMBER\n";

cout<<"PRESS 2 FOR FIBONACCI SERIES\n";

cout<<"PRESS 3 FOR SWAP OF TWO NUMBER WITHOUT USING THIRD ELEMENT\n";

cout<<"PRESS 4 FOR MAX ELEMENT IN ARRAY\n";

cout<<"PRESS 5 FOR SECOND MAX ELEMENT IN ARRAY\n";

cout<<"PRESS 6 FOR EXIT\n\n";

cout<<"ENTER YOUR CHOISE:";

cin>>i;

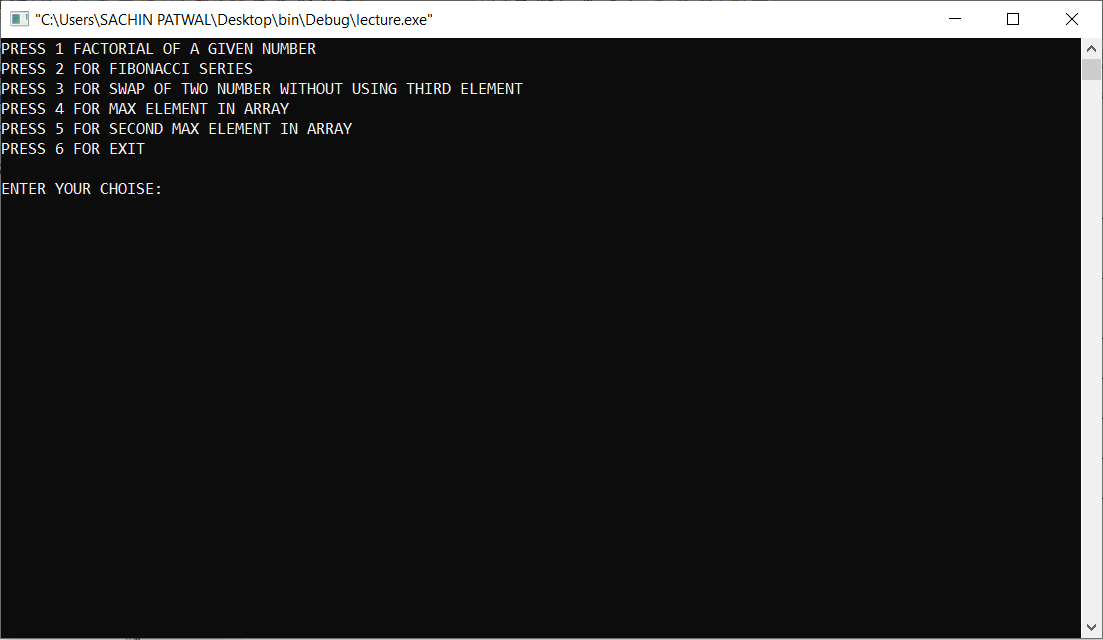


Fig No.1.1: Welcome ScreenFig

switch(i)

{ case 1: cls();

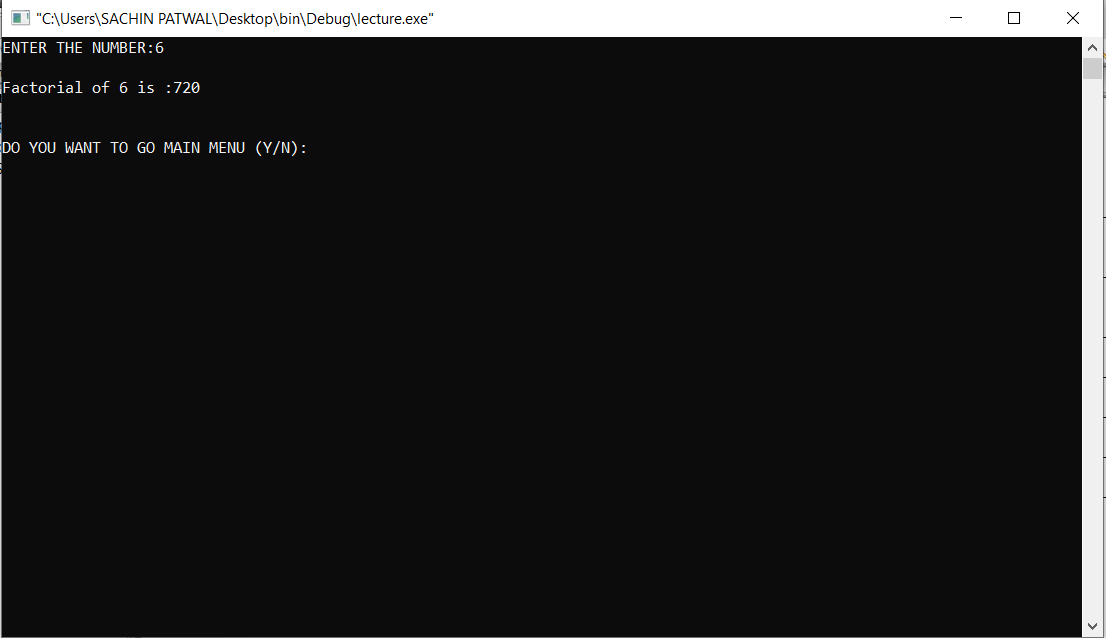
cout<<"ENTER THE NUMBER:";

cin>>n;

f=factorial(n);

cout<<"\nFactorial of "<<n<<" is :"<<f<<endl;

break;



No. 1.2: Factorial of a number

case 2: cls();

cout<<"ENTER NUMBER OF TERMS YOU WANT TO ENTER:";

cin>>fi;

fibbo(fi);

break;

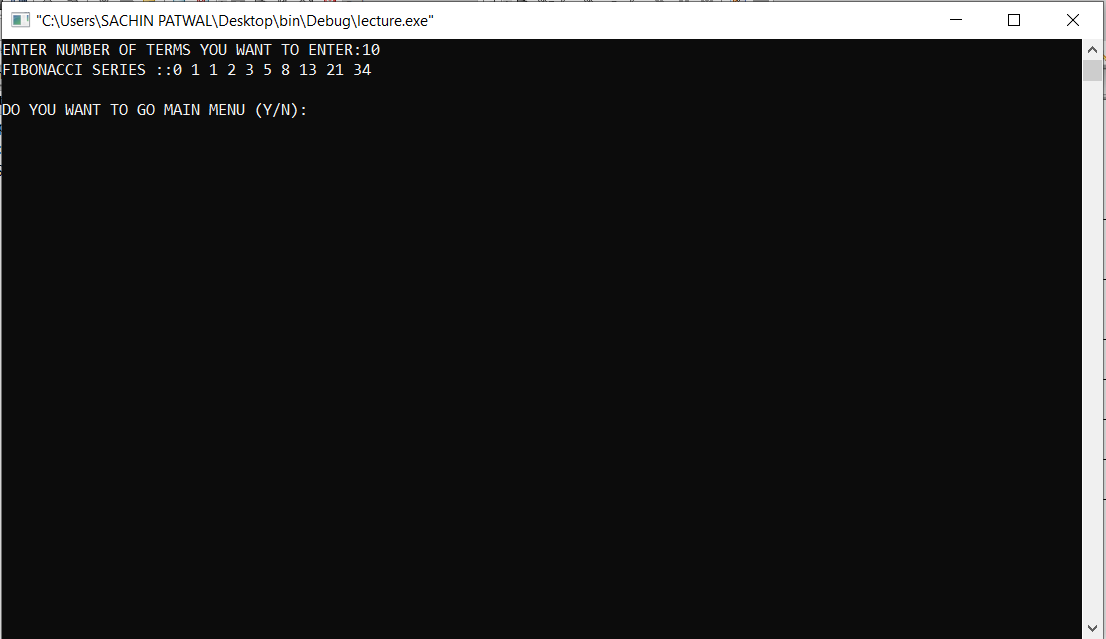


Fig No. 1.3: Fibonacci Series

case 3: cls();

cout<<"ENTER ANY TWO NUMBERS\n";

cin>>a>>b;

cout<<"\nNUMBER BEFORE SWAP:"<<a<<" "<<b;

swapi(a,b);

cout<<"\nNUMBER AFTER SWAP:"<<a<<" "<<b;

break;

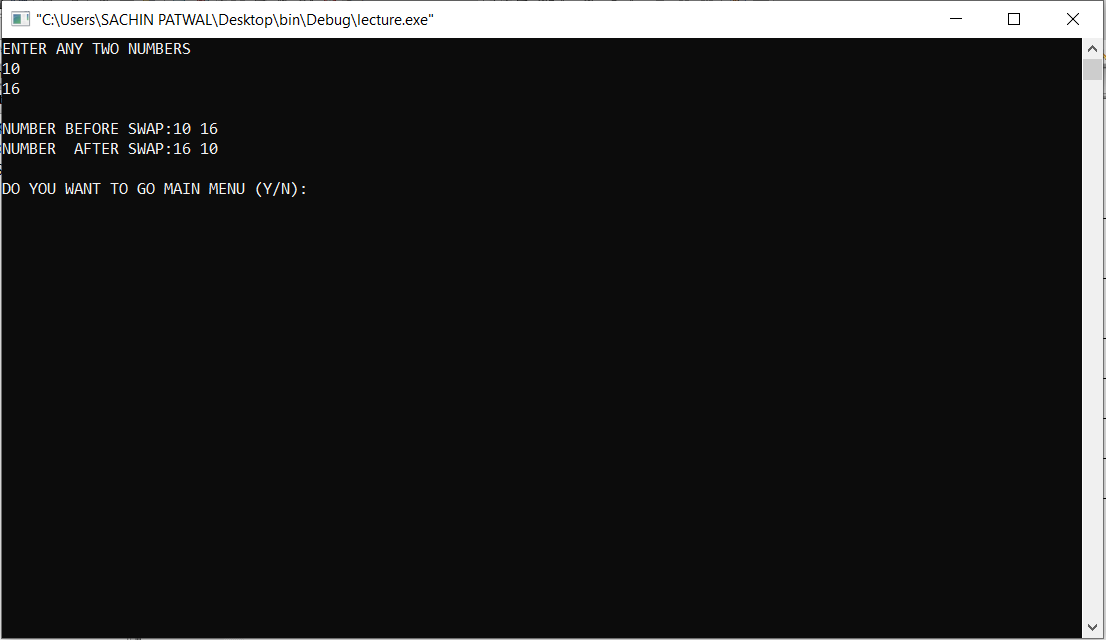


Fig No. 1.4: Swapping two numbers

case 4: cls();

cout<<"ENTER THE SIZE OF ARRAY:";

cin>>n1;

cout<<"\nENTER THE ARRAY:\n";

for(int i=0;i<n1;i++)

cin>>x[i];

maximum(x,n1);

break;

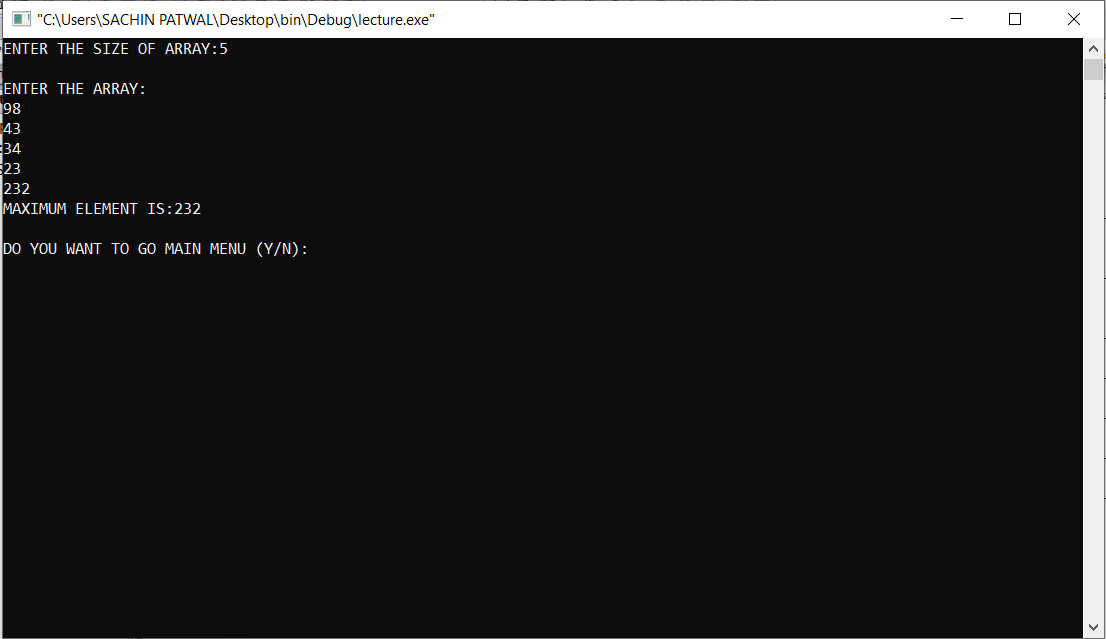


Fig No. 1.5: Largest number in an array

case 5: cls();

cout<<"ENTER THE SIZE OF ARRAY:";

cin>>n2;

cout<<"\nENTER THE ARRAY:\n";

for(int i=0;i<n2;i++)

cin>>y[i];

second\_maximum(y,n2);

break;

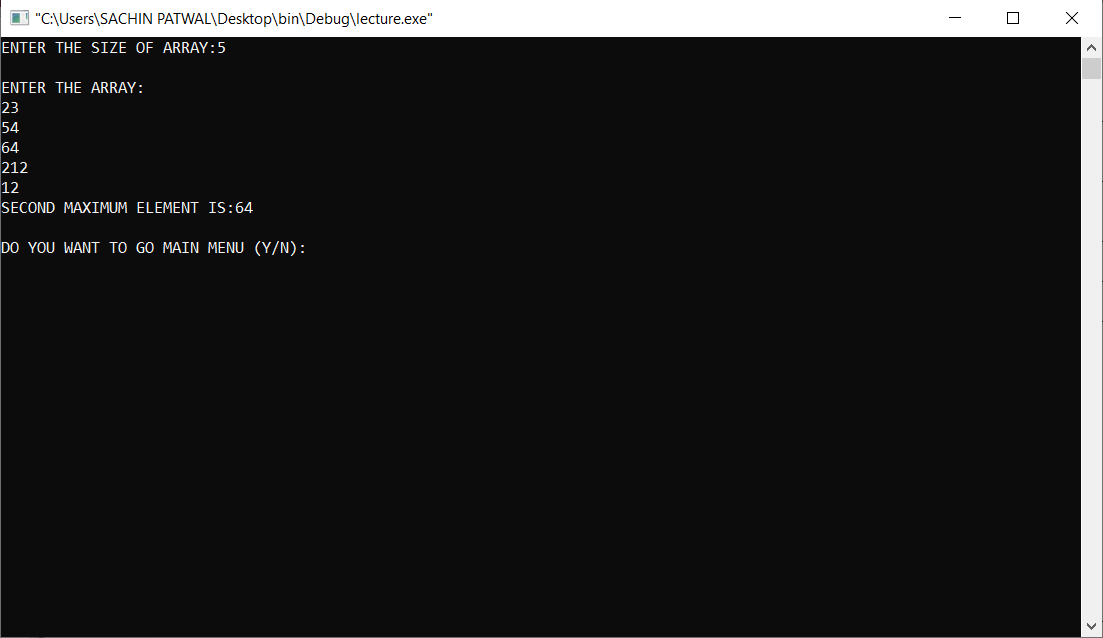


Fig No. 1.6: Largest and Second largest number in an array

case 6: cls();

cout<<"PROGRAM HAS BEEN SUCCESFULLY CLOSED\n\n";

exit(0);

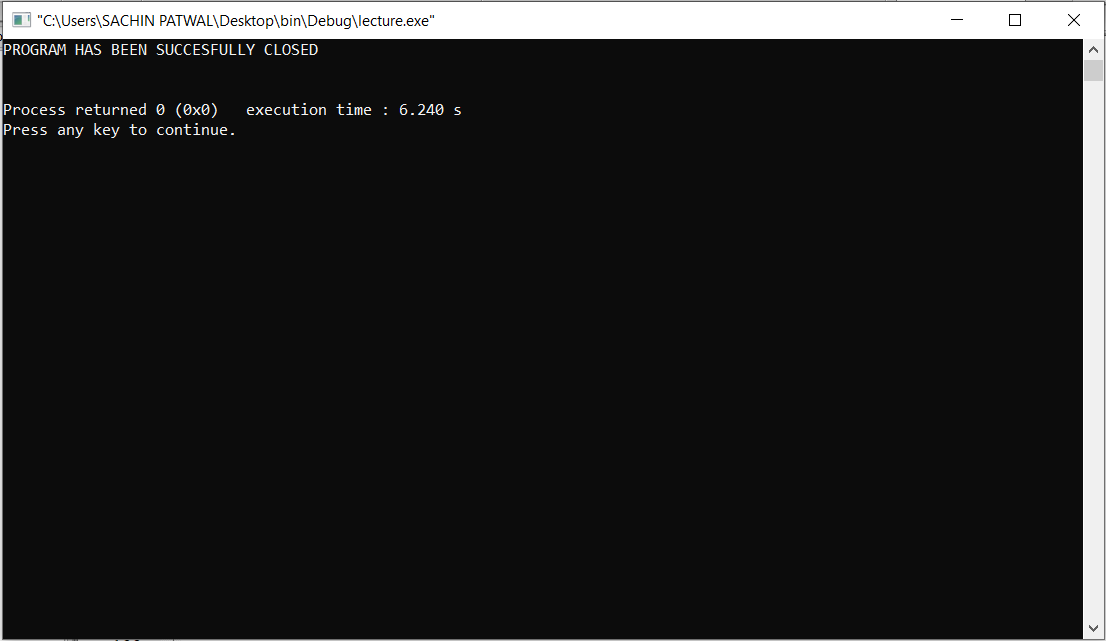


Fig No.1.7: Exit

default: cls();

cout<<"YOU HAVE ENTERED AN INVALID CHOISE\n\n";

break;

} r++;

cout<<"\n\nDO YOU WANT TO GO MAIN MENU (Y/N):";

cin>>c;

} return 0;

}

**Experiment No. 2**

Write a menu-based program that performs following operations:

1. Find the sum of array elements
2. Insert a new an element in given location
3. Delete an element from the given location
4. Search an element using linear search algorithm
5. Search an element using binary search algorithm
6. Sort the elements of an array using bubble sort algorithm
7. Exit the program

**SOURCE CODE**

#include <iostream>

#include<stdlib.h>

#include<stdio.h>

using namespace std;

void cls()

{

system("cls");

}

void in( int a[],int n)

{

cout<<"ENTER THE ARRAY:\n";

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

cin>>a[i];

}

void out( int a[],int n)

{

cout<<"\nARRAY IS:";

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

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

}

void sum(int a[],int n)

{

int sum=0;

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

sum=sum+a[i];

cout<<"\nSUM OF ARRAY IS:"<<sum;

}

void insertion(int a[],int &n,int l)

{

int item,n1=n+1;

cout<<"\nENTER THE ELEMENT U WANT TO INSERT:";

cin>>item;

for(int i=n;i>=l;i--)

a[i+1]=a[i];

a[l]=item;

n++;

}

void deletion(int a[],int &n,int l)

{

for(int i=l;i<n;i++)

a[i]=a[i+1];

n--;

}

void lsearch(int a[],int n)

{ int s,loc,flag=0;

cout<<"ENTER THE ELEMENT YOU WANT TO SEARCH:";

cin>>s;

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

{if(a[i]==s)

{

flag=1;

loc=i+1;

} }

if(flag==1)

cout<<"\nITEM FOUND ON LOCATION:"<<loc;

else

cout<<"\nITEM NOT FOUND";

}

void bisearch(int a[],int n)

{ int s,loc,flag=0,l=0,u=n-1,mid=int((u+l)/2);

cout<<"ENTER THE ELEMENT YOU WANT TO SEARCH:";

cin>>s;

while(l<=u)

{ if(a[mid]==s)

{flag=1;

loc=mid+1;

break;

}

else if(s<a[mid])

{

u=mid-1;

mid=((u+l)/2);

} else

{

l=mid+1;

mid=((u+l)/2);

} }

if(flag==1)

cout<<"\nITEM FOUND ON LOCATION:"<<loc;

else

cout<<"\nITEM NOT FOUND";

}

void sorting (int a[],int &n)

{ int t;

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

{

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

{

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

{ t=a[j+1];

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

a[j]=t;

}

}

}

}

int main()

{

int i,n,r=0,a[100],b[100],n1,loc1,n2,d[100],loc2,j,n3,n4,e[100],f[100],g[100];

char c='y';

while(c=='y'||c=='Y')

{ if(r!=0)

cls();

cout<<"PRESS 1 FOR SUM OF ARRAY\n";

cout<<"PRESS 2 FOR INSERT A NEW ELEMENT AT A GIVEN LOCATION IN ARRAY\n";

cout<<"PRESS 3 FOR DELETE AN ELEMENT FORM A GIVEN LOCATION IN ARRAY\n";

cout<<"PRESS 4 FOR SEARCH IN ARRAY \n";

cout<<"PRESS 5 FOR BUBBLE SORT\n";

cout<<"PRESS 6 FOR EXIT\n\n";

cout<<"ENTER YOUR CHOISE:";

cin>>i;

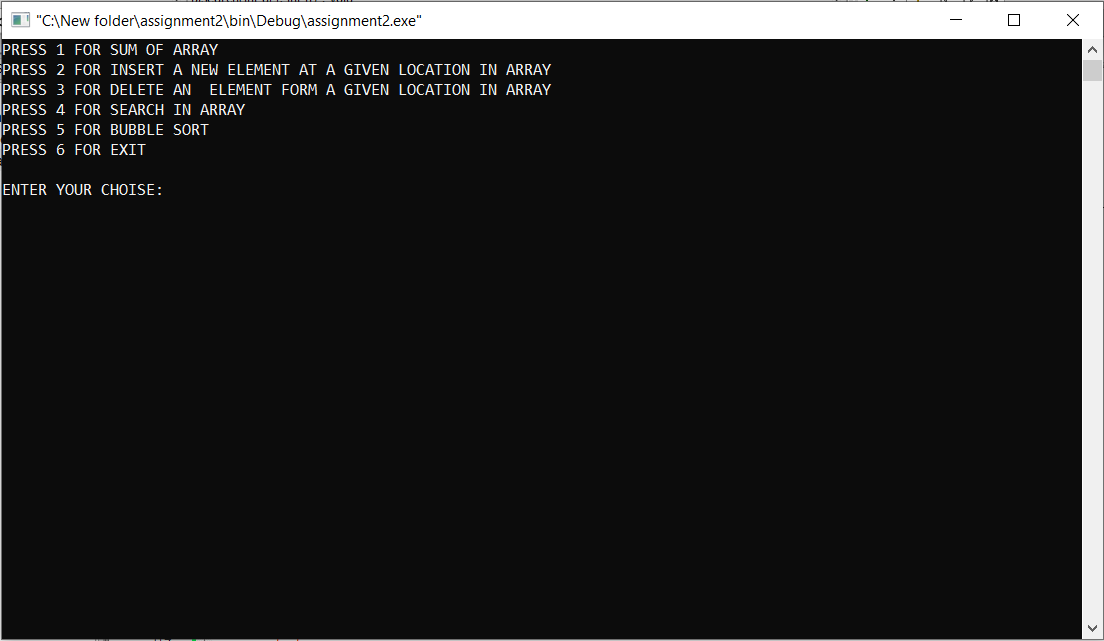


Fig No 2.1: Welcome Screen

switch(i)

{

case 1: cls();

cout<<"ENTER THE SIZE OF ARRAY:";

cin>>n;

in(a,n);

sum(a,n); break;

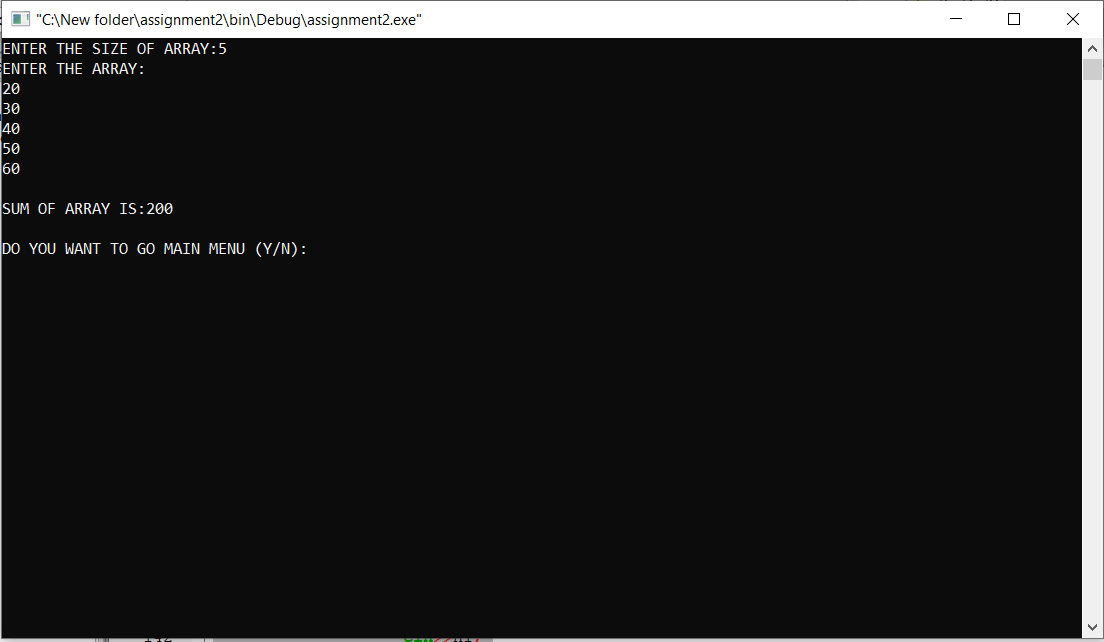


Fig No 2.2: Sum of the array elements

case 2: cls();

cout<<"ENTER THE SIZE OF ARRAY:";

cin>>n1;

in(b,n1);

cout<<"\nENTER THE LOCATION WHERE YOU WANT TO ENTER THE ELEMENT:";

cin>>loc1;

insertion(b,n1,loc1);

out(b,n1);

break;

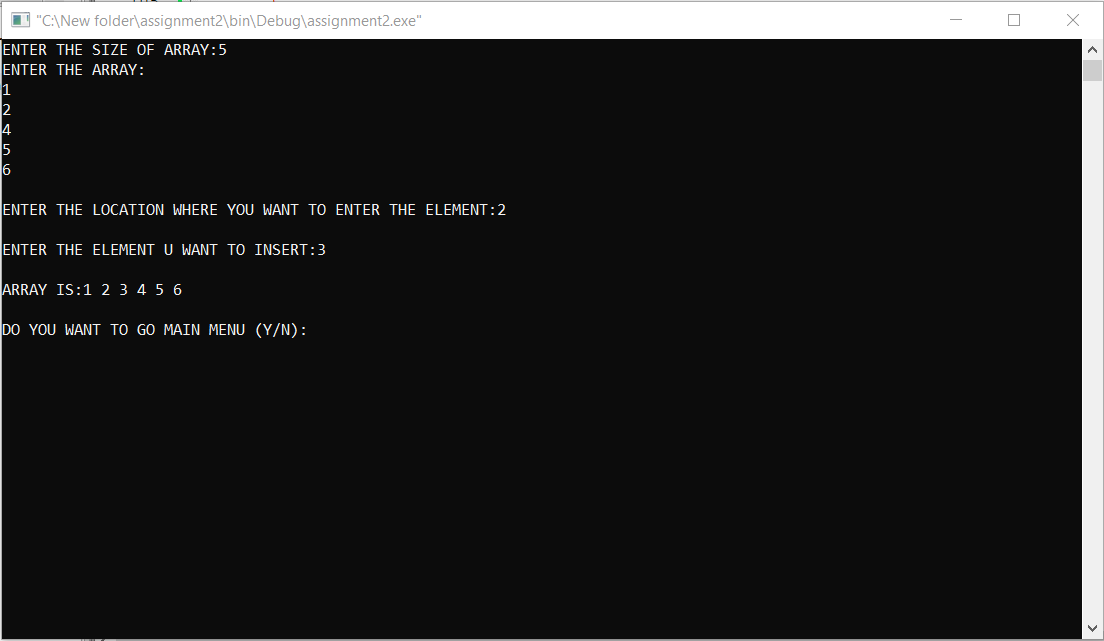
****

Fig No. 2.3: Insert an element in the given location

case 3: cls();

cout<<"ENTER THE SIZE OF ARRAY:";

cin>>n2;

in(d,n2);

cout<<"ENTER THE LOCATION FROM WHERE YOU WANT TO DELETE THE ELEMENT:";

cin>>loc2;

deletion(d,n2,loc2);

cout<<"\nAFTER DELETION\n";

out(d,n2);

break;

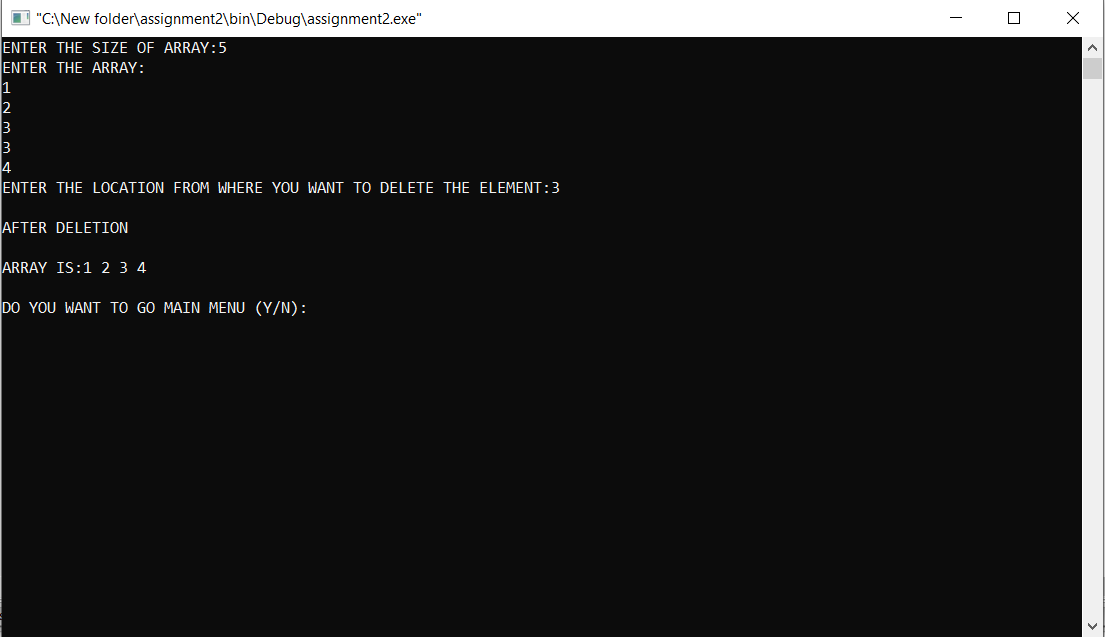


Fig No. 2.4: Delete an element from the given location

case 4: cls();

cout<<"PRESS 1 FOR LINEAR SEARCH";

cout<<"\nPRESS 2 FOR BINARY SEARCH";

cout<<"\n\nENTER YOUR CHOISE:";

cin>>j;

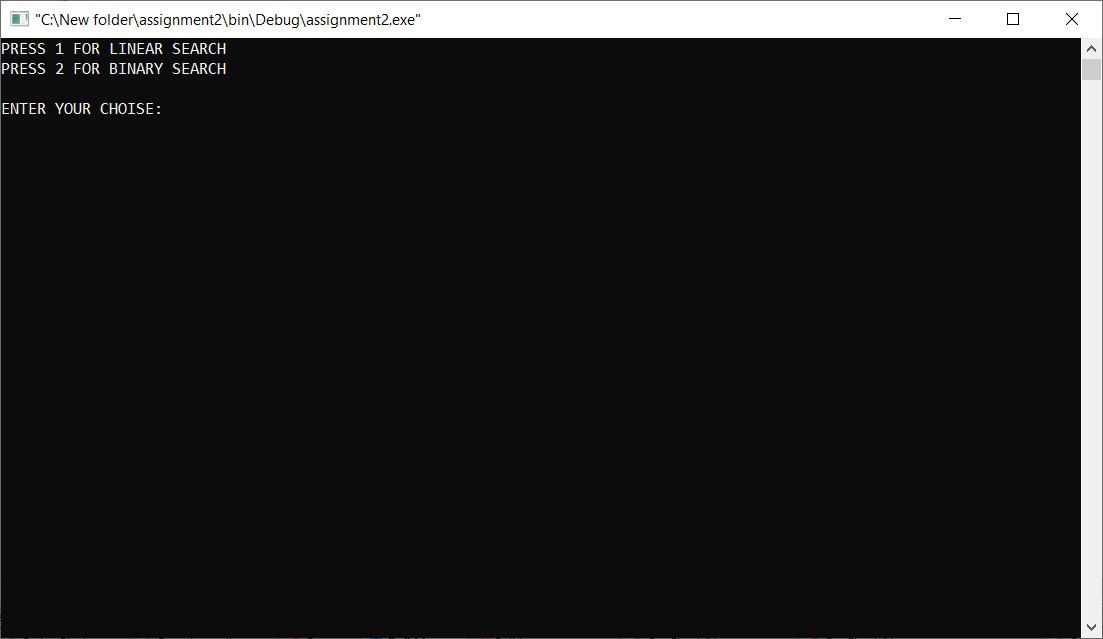
****

Fig No. 2.5:Searching

switch(j)

{ case 1: cls();

cout<<"ENTER THE SIZE OF ARRAY:";

cin>>n3;

in(e,n3);

lsearch(e,n3);

break;

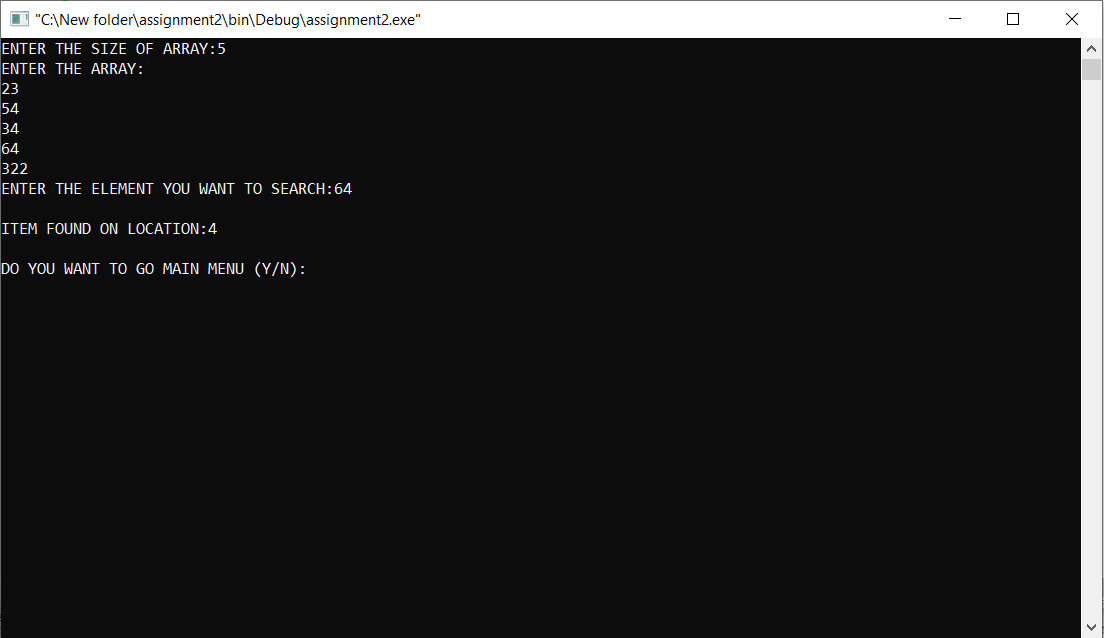
****

Fig No. 2.6: Search an element using linear search algorithm

case 2: cls();

cout<<"PLEASE ENTER THE IN SHORTED ORDER \n\n";

cout<<"ENTER THE SIZE OF ARRAY:";

cin>>n3;

in(f,n3);

bisearch(f,n3);

break; }

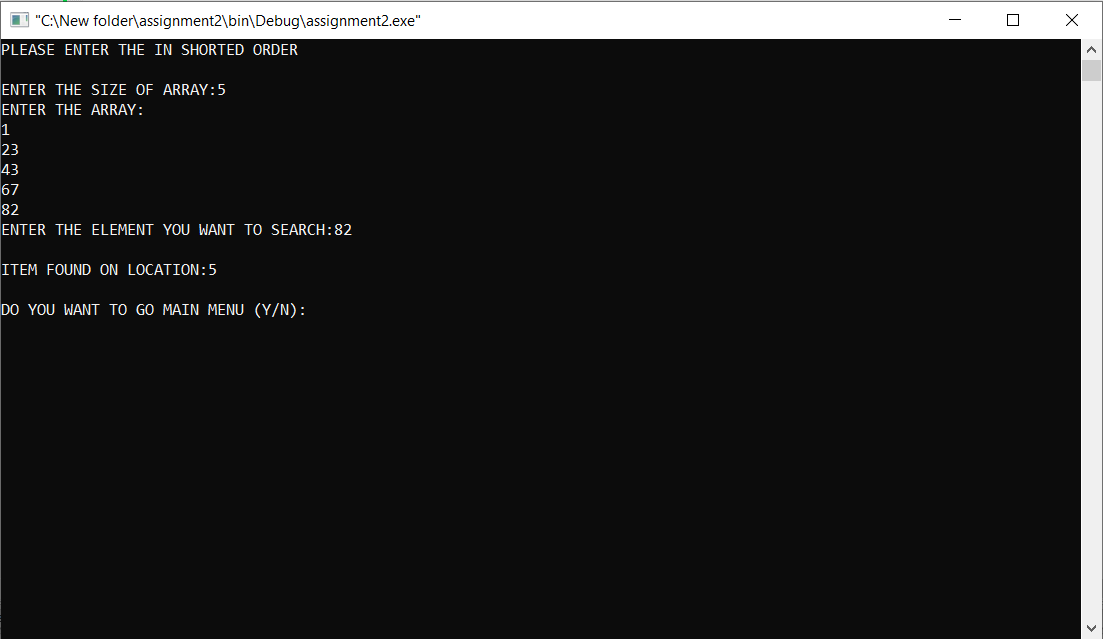


Fig No. 2.7: Search an element using binary search algorithm

case 5: cls();

cout<<"ENTER THE SIZE OF ARRAY:";

cin>>n4;

in(g,n4);

cout<<"\nARRAY BEFORE SHORTING";

out(g,n4);

sorting(g,n4);

cout<<"\nARRAY AFTER SHORTING";

out(g,n4);

break;

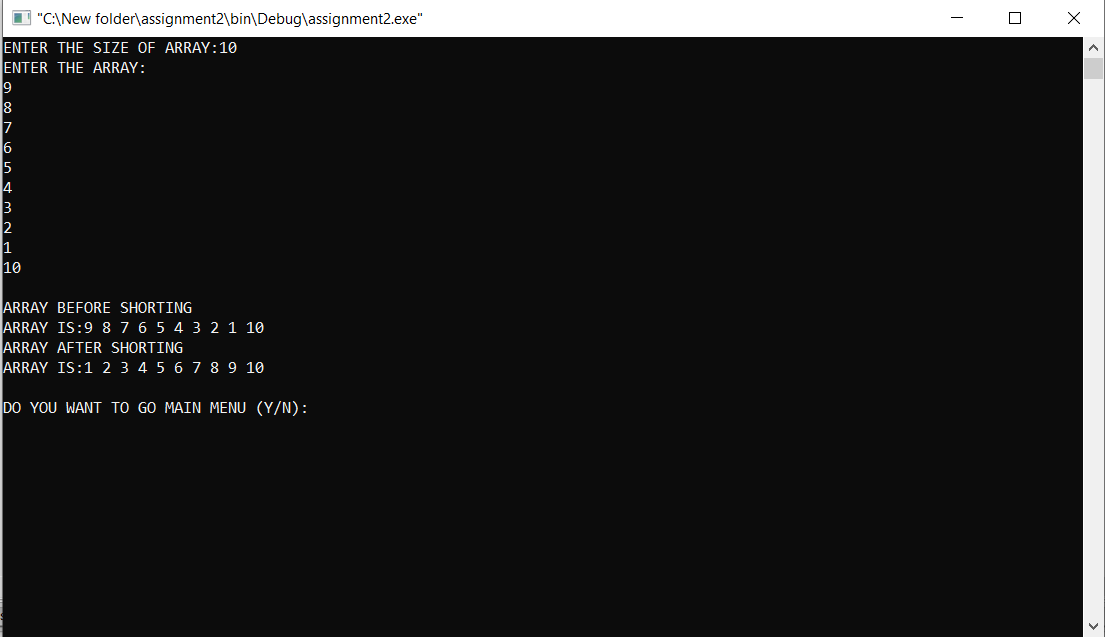


Fig No. 2.8: Sort the elements of an array using bubble sort algorithm

case 6: cls();

cout<<"PROGRAM HAS BEEN SUCCESFULLY CLOSED\n\n";

break;

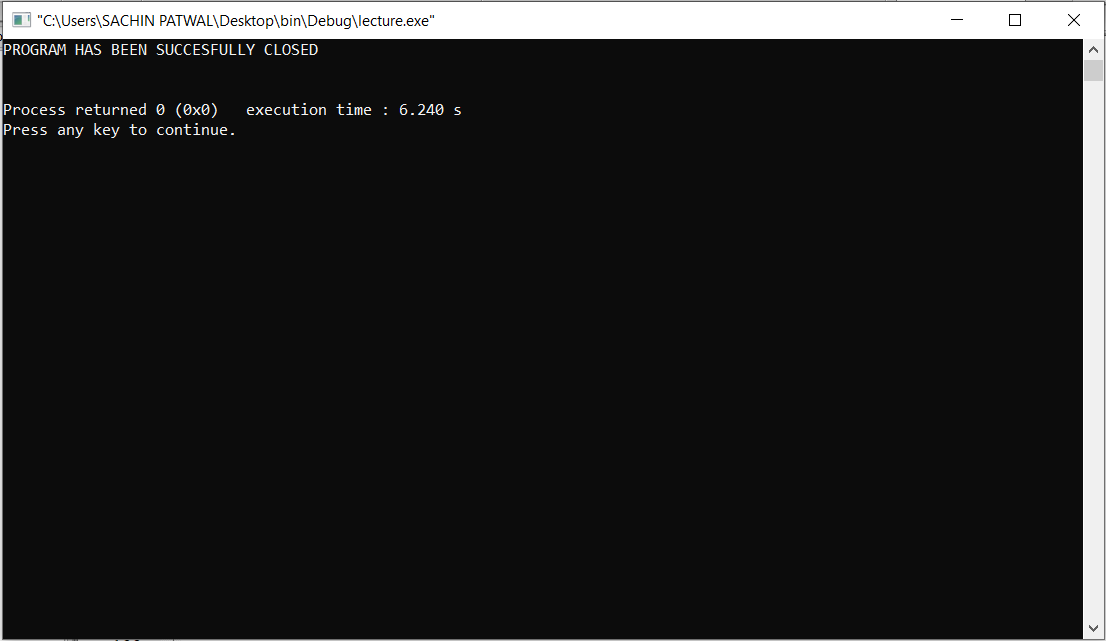


Fig No.2.9: Exit

default: cls();

cout<<"YOU HAVE ENTERED AN INVALID CHOISE\n\n";

break;

} r++;

cout<<"\n\nDO YOU WANT TO GO MAIN MENU (Y/N):";

cin>>c; } return 0;

}