**CSC221: DATA STRUCTURES & ALGORITHMS**

**BSCS 3*B***

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | LAB | | **03** | | Implementation of searching algorithms over an Array based List   * Linear Search * Binary Search |



**Submitted By:**

* Ahsan Ghaffar
* Reg: 48411
* BS(CS)\_3B

**Submitted to:**

* Miss Ambreen Akram(AA)

**Submission Date:**

[05/03/2018]

**DEPARTMENT OF COMPUTER SCIENCE**

**BAHRIA UNIVERSITY, KARACHI CAMPUS**

**TASK\_01:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Translate the matrix multiplication algorithm into a program which finds the product C of an n x m matrix A and p x n matrix B. Test the program using   |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **A** = | |  |  |  | | --- | --- | --- | | | 4 | -3 | 5 | | | | |  | | | | | 6 | 1 | -2 | | |  | |  |  |  | | **B** = | |  |  |  |  | | --- | --- | --- | --- | | | 2 | 3 | -7 | -3 | | | | 5 | -1 | 6 | 2 | | | | 0 | 3 | -2 | 1 | | |  | |

**SOURCE CODE:**

#include<iostream>

#include<iomanip>

using namespace std;

int main()

{

//Declaring Variable:

int i,j,k;

int const r1=2,c1=3,r2=3,c2=4;

//Declaring Arrays:

int a[r1][c1],b[r2][c2],c[r1][c2],sum=0;

//The Whole Program is Skipped If Column1 is Not Equal to Row2:

if(c1!=r2)

{

cout<<"Matrix Multiplication is not Possible"<<endl;

return 0;

}

cout<<"\nEnter Values Of Matrix A:"<<"\n\n";

//Taking Input Of Matrix A Values from User:

for(i=0;i<r1;i++)

{

for(j=0;j<c1;j++)

{

cout<<"Enter["<<i<<"]["<<j<<"] :";

cin>>a[i][j];

}

}

cout<<"\nEnter Values Of Matrix B:"<<"\n\n";

//Taking Input Of Matrix B Values from User:

for(i=0;i<r2;i++)

{

for(j=0;j<c2;j++)

{

cout<<"Enter["<<i<<"]["<<j<<"] :";

cin>>b[i][j];

}

}

/\*Apply Nested For loop To calculate Multiplication Of

Matrix A & B And Store It In Matrix C:\*/

for(i=0;i<r1;i++)

{

for(j=0;j<c2;j++)

{

for(k=0;k<r2;k++)

{

sum+=a[i][k]\*b[k][j];

}

c[i][j]=sum;

sum=0;

}

}

cout<<"\n Matrix A"<<endl<<"\n";

//Prints Matrix A:

for(i=0;i<r1;i++)

{

cout<<"\t";

for(j=0;j<c1;j++)

{

cout<<setw(4)<<a[i][j];

}

cout<<endl;

}

cout<<"\n Matrix B"<<endl<<"\n";

//Prints Matrix B:

for(i=0;i<r2;i++)

{

cout<<"\t";

for(j=0;j<c2;j++)

{

cout<<setw(4)<<b[i][j];

}

cout<<endl;

}

cout<<"\nMultiplication Of Matrix A & B:"<<endl;

cout<<"\n Matrix C"<<endl<<"\n";

//Prints Multiplication Of Matrix A & Matrix B:

for(i=0;i<r1;i++)

{

cout<<"\t";

for(j=0;j<c2;j++)

{

cout<<setw(5)<<c[i][j];

}

cout<<endl;

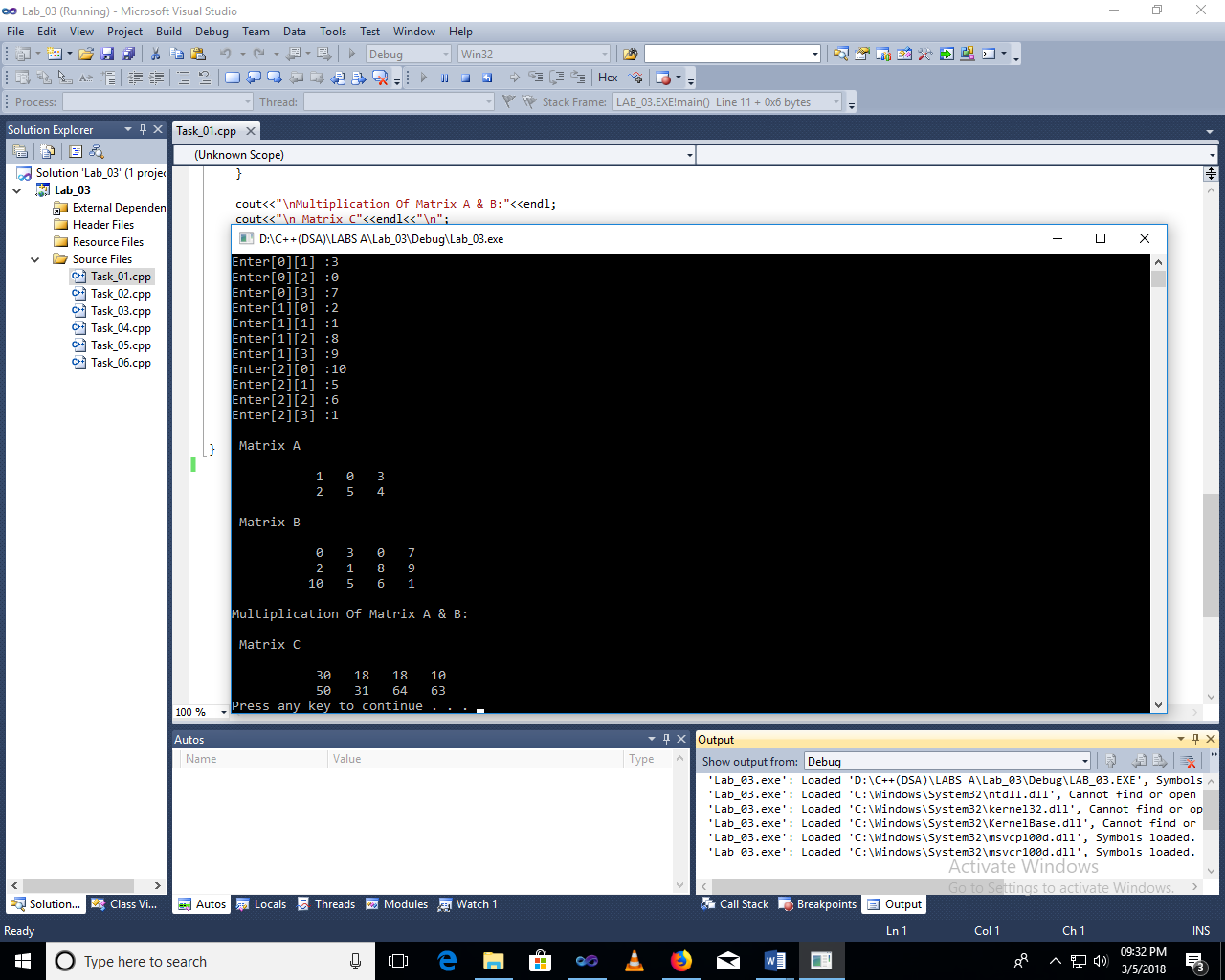
}

system("pause");

return 0;

}

**SCREENSHOT:**



**TASK\_02:**

|  |  |
| --- | --- |
| 2 | Suppose A and B are n-elements vector array in memory and X an Y are scalars. Write a program to find.  a) XA + YB  b) A . B  Test the program using A= (16, -6,7), B=(4,2,-3), X= 2, Y= -5 |

**SOURCE CODE:**

#include<iostream>

using namespace std;

void Part\_A(int A[],int B[],int X,int Y);

void Part\_B(int A[],int B[]);

int main()

{

int X=2 , Y=-5 , A[]={16,-6,7} , B[]={4,2,-3};

Part\_A(A,B,X,Y);

Part\_B(A,B);

system("pause");

return 0;

}

void Part\_A(int A[],int B[],int X,int Y)

{

int temp[3]={0};

for(int i=0;i<3;i++)

{

A[i]\*=X;

B[i]\*=Y;

}

cout<<"The [Array(A)\*X] answer is:"<<endl;

for(int i=0;i<3;i++)

cout<<A[i]<<endl;

cout<<"\nThe [Array(B)\*Y] answer is:"<<endl;

for(int i=0;i<3;i++)

cout<<B[i]<<endl;

for(int i=0;i<3;i++)

{

temp[i]+=A[i];

temp[i]+=B[i];

}

cout<<"\n\n[(Array(A)\*X)+(Array(B)\*Y)] answer is: "<<endl;

for(int i=0;i<3;i++)

cout<<temp[i]<<endl;

}

void Part\_B(int A[],int B[])

{

int temp2[3]={0};

for(int i=0;i<3;i++)

{

A[i]\*=B[i];

temp2[i]+=A[i];

}

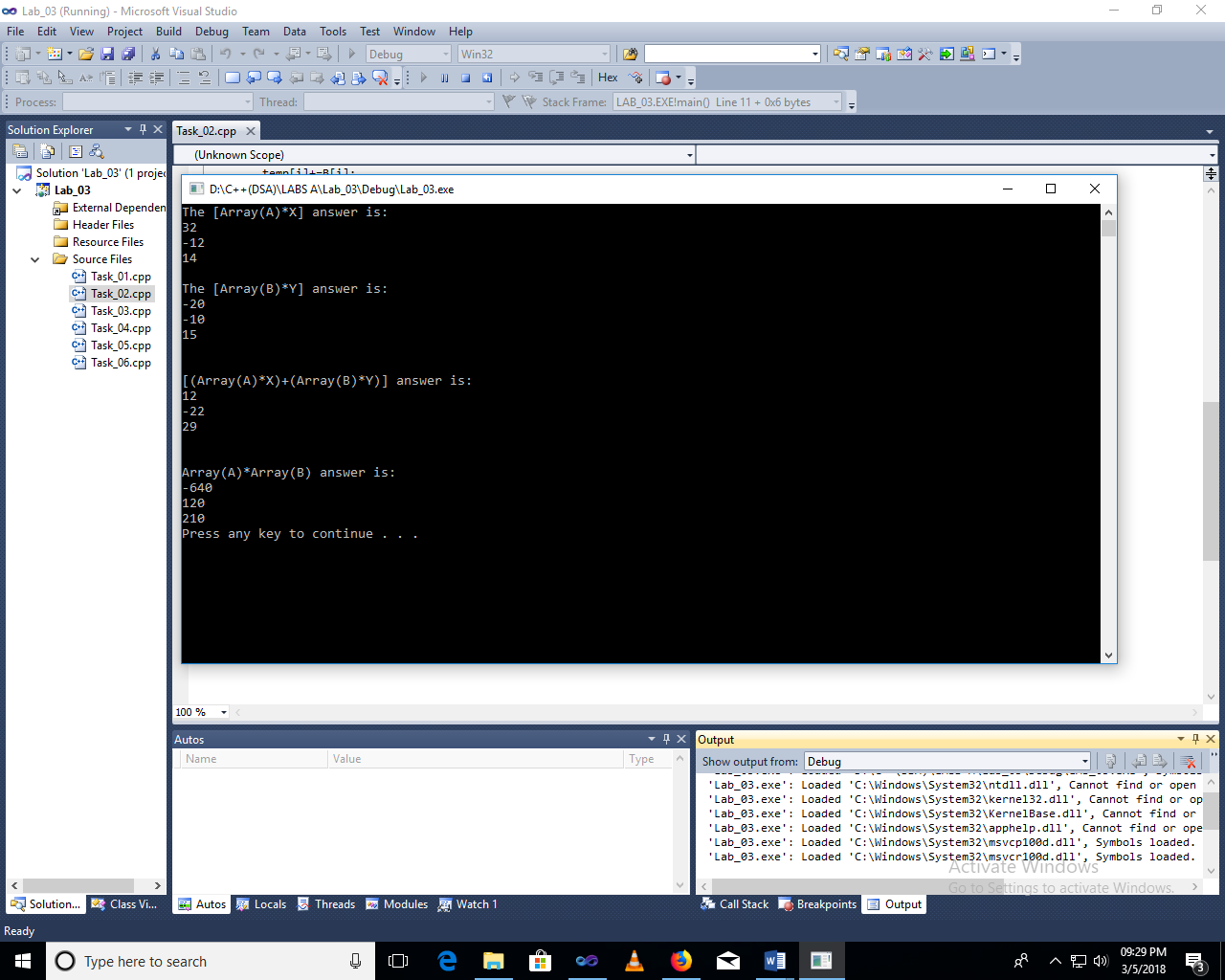
cout<<"\n\nArray(A)\*Array(B) answer is: "<<endl;

for(int i=0;i<3;i++)

cout<<temp2[i]<<endl;

}

**SCREENSHOT:**



**TASK\_03:**

|  |  |
| --- | --- |
| 3 | Translate the linear search algorithm into a program which either find the location LOC where ITEM appears in ARRAY or return LOC=0 |

**SOURCE CODE:**

#include<iostream>

using namespace std;

int main()

{

int a,arr1[5],loc;

cout<<"Enter The arr1 values"<<endl;

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

{

cin>>arr1[i];

}

cout<<"\nEnter the value to find it's location from the above array:"<<endl;

cin>>a;

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

{

if(arr1[i]==a)

{

loc=i;

cout<<"\nLocation of the Resultant Value ("<<a<<") is: "<<loc<<endl;

break;

}

}

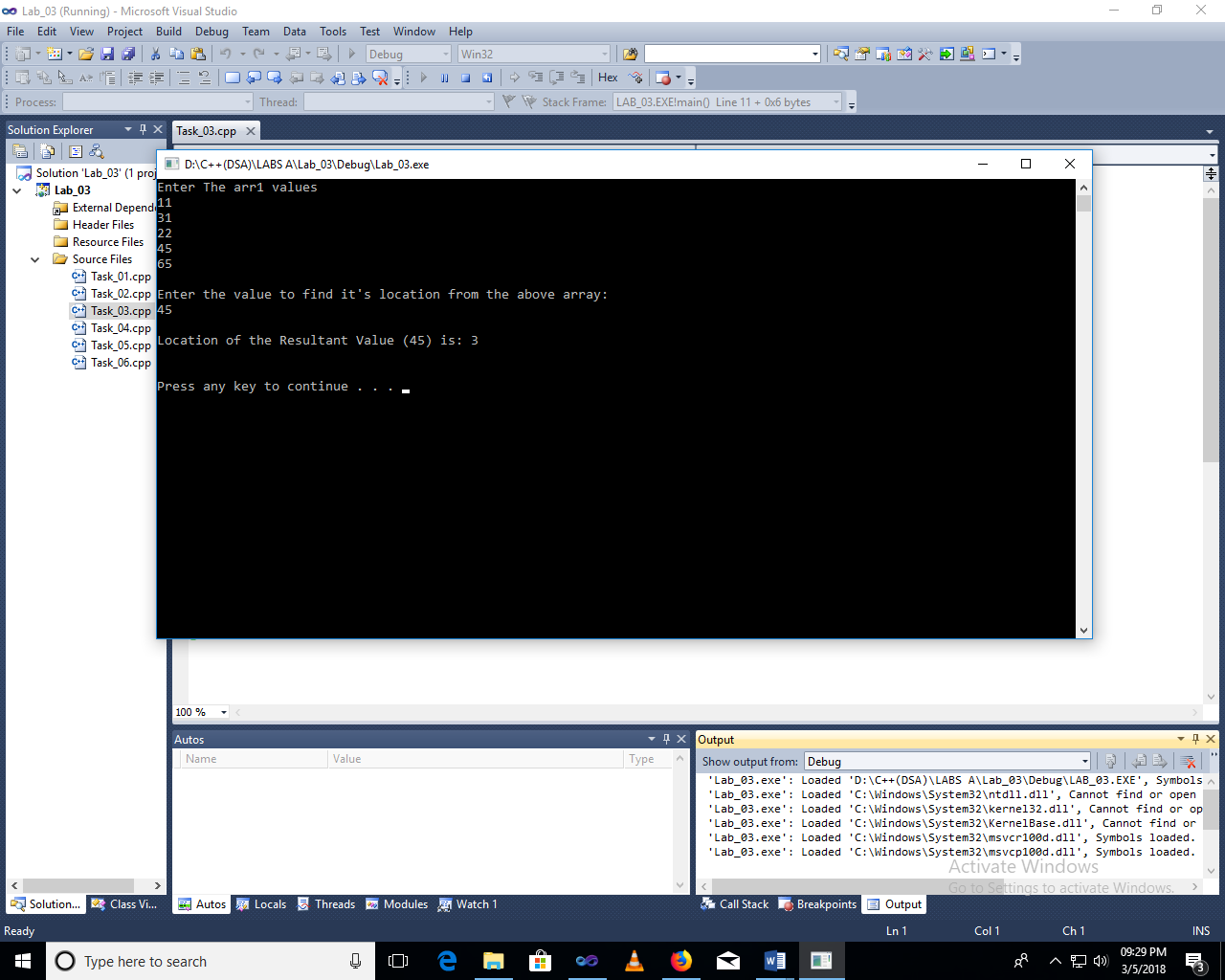
cout<<"\n\n";

system("pause");

return 0;

}

**SCREENSHOT:**



**TASK\_04:**

|  |  |
| --- | --- |
| 4 | Translate binary search and insertion algorithm into a program which finds either the location LOC where ITEM appears in ARRAY or the location LOC where ITEM should be inserted into ARRAY. (For object 3, 4, and 5 take an array of 10 elements given below) |

**SOURCE CODE:**

#include<iostream>

using namespace std;

void Binary\_ser(int a[],int size,int value);

int main()

{

int const size=10;

int a[size]={11,22,33,36,45,52,57,60,64,78},b[size]={0},temp=0,x=36,y=45,z=52;

cout<<"Array Elements:"<<endl;

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

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

cout<<"\n\n";

cout<<"\nThe Value at the Location '";

Binary\_ser(a,size,x);

cout<<"' is: "<<x;

cout<<"\nThe Value at the Location '";

Binary\_ser(a,size,y);

cout<<"' is: "<<y;

cout<<"\nThe Value at the Location '";

Binary\_ser(a,size,z);

cout<<"' is: "<<z<<"\n\n";

system("pause");

return 0;

}

void Binary\_ser(int a[],int size,int value)

{

int counter=0,lb=0,ub=size-1,mid,position=0;

bool found=false;

while(!found && lb<=ub)

{

mid=(lb+ub)/2;

if(a[mid]==value)

{

found=true;

position=mid;

}

else if (a[mid]>value)

ub=mid-1;

else

lb=mid+1;

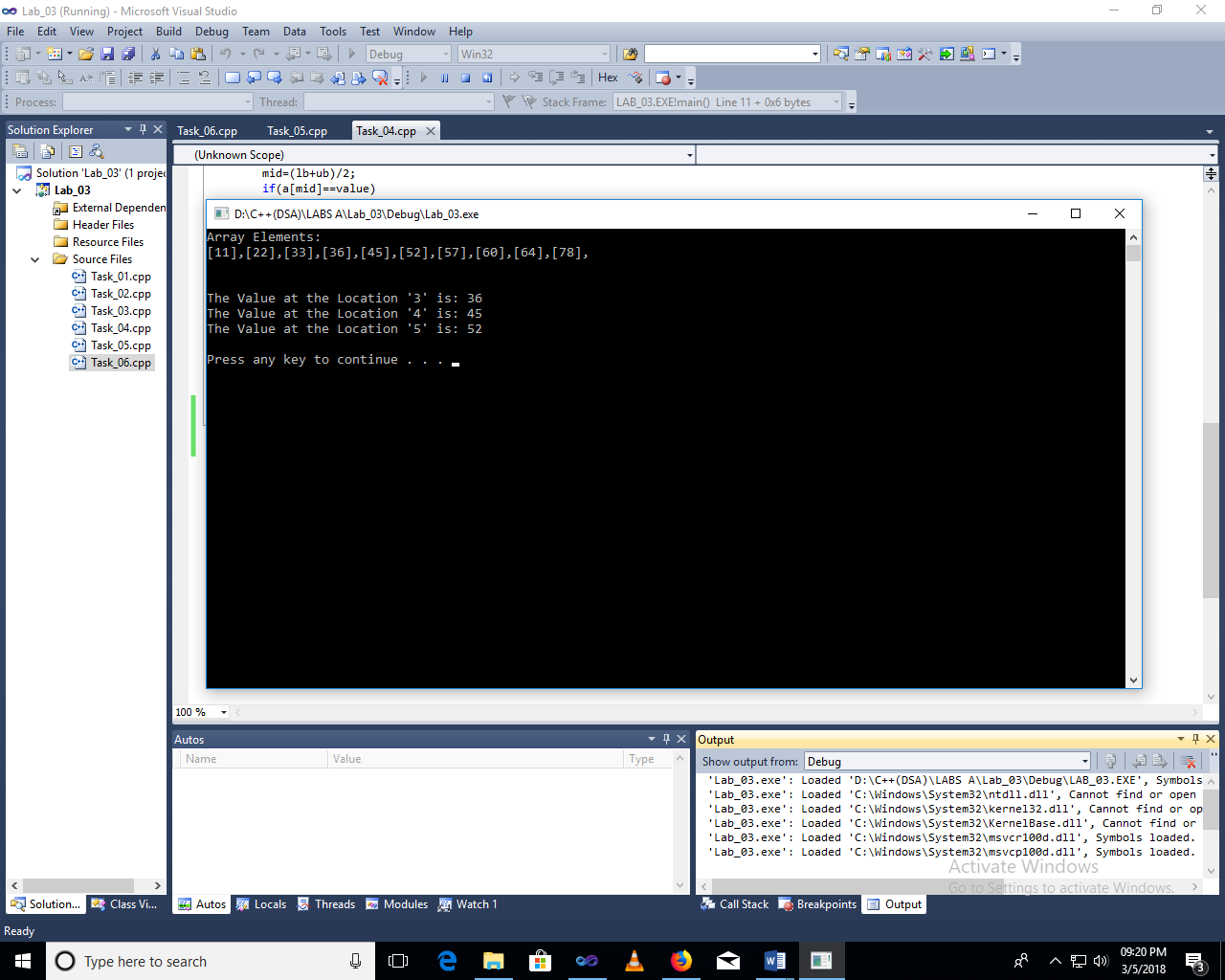
counter++;

}

cout<<position;

}

**SCREENSHOT:**



**TASK\_05:**

|  |  |
| --- | --- |
| 5 | Write a program which uses Binary Search to search the elements 52,and 33 and print it. |

**SOURCE CODE:**

#include<iostream>

using namespace std;

void Binary\_ser(int a[],int size,int value);

void Sort(int a[],int size);

int main()

{

int const size=10;

int a[size]={10,4,11,25,19,33,77,52,1,63},b[size]={0},temp=0,x=52,y=33;

cout<<"UnSorted Array:"<<endl;

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

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

cout<<"\n\n";

Sort(a,size);

cout<<"\nThe Position of the value '"<<x<<"' is: ";

Binary\_ser(a,size,x);

cout<<"The Position of the value '"<<y<<"' is: ";

Binary\_ser(a,size,y);

system("pause");

return 0;

}

void Sort(int a[],int size)

{

int temp=0;

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

{

for(int j=i;j>=1;j--)

{

if(a[j]<a[j-1])

{

temp=a[j];

a[j]=a[j-1];

a[j-1]=temp;

}

}

}

cout<<"\nSorted Array:"<<endl;

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

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

cout<<"\n\n";

}

void Binary\_ser(int a[],int size,int value)

{

int counter=0,lb=0,ub=size-1,mid,position=0;

bool found=false;

while(!found && lb<=ub)

{

mid=(lb+ub)/2;

if(a[mid]==value)

{

found=true;

position=mid;

}

else if (a[mid]>value)

ub=mid-1;

else

lb=mid+1;

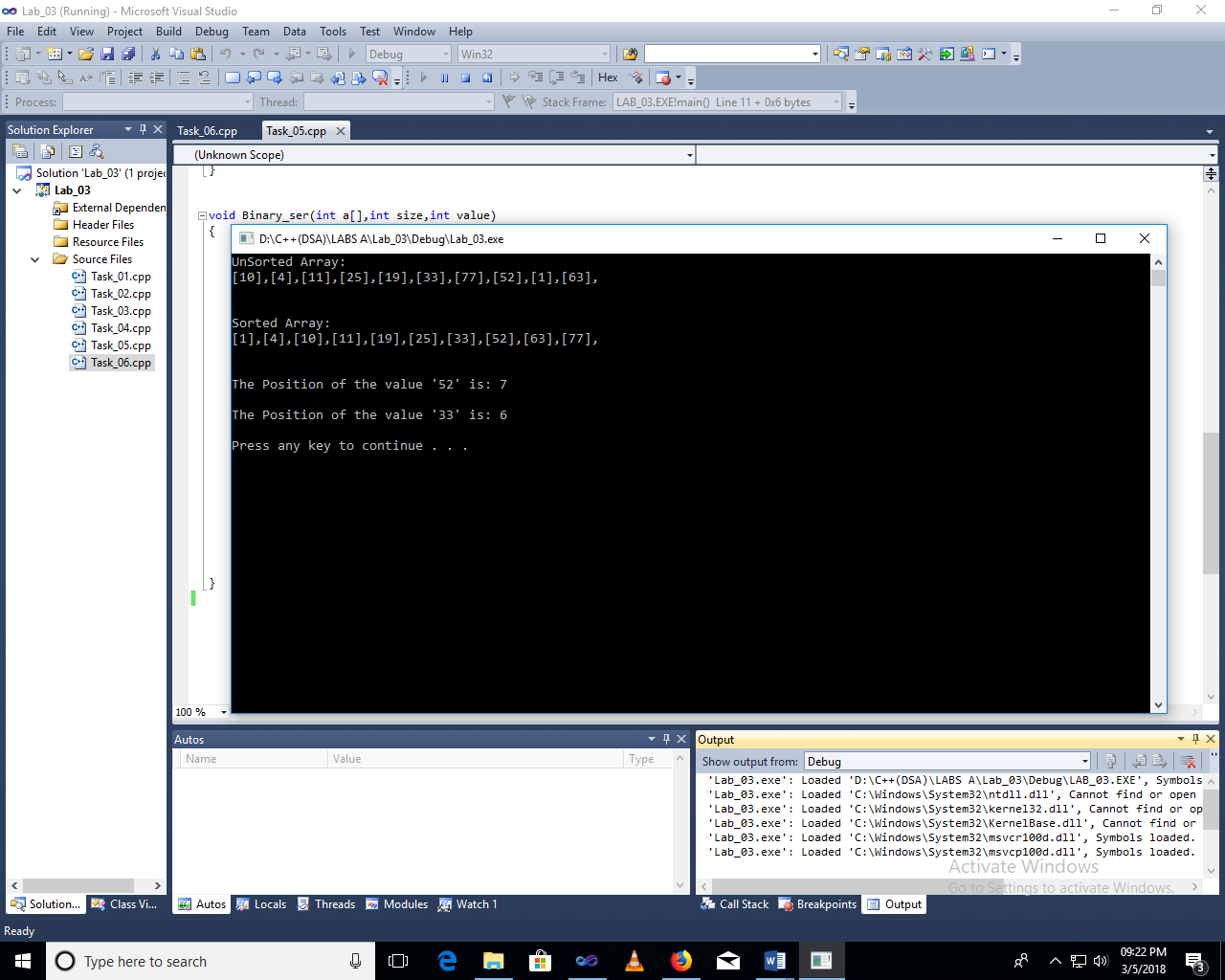
counter++;

}

cout<<position<<endl<<"\n";

}

**SCREENSHOT:**



**TASK\_06:**

|  |  |
| --- | --- |
| 6 | Write a program which uses Binary Search to search the elements 45,and 78 and delete it. NUMBERS=[11,22,33,36,45,52,57,60,64,78] |

**SOURCE CODE:**

#include<iostream>

using namespace std;

int Binary\_ser(int a[],int size,int value);

void delete1(int a[],int size,int loc);

int main()

{

int const size=10;

int a[size]={11,22,33,36,45,52,57,60,64,78},b[size]={0},x=45,y=78,temp1=0,temp2=0;

cout<<"Array Elements:"<<endl;

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

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

cout<<"\n\n";

temp1=Binary\_ser(a,size,x);

cout<<"\nThe Position of the value '"<<x<<"' is: "<<temp1;

temp2=Binary\_ser(a,size,y);

cout<<"\n\nThe Position of the value '"<<y<<"' is: "<<temp2;

delete1(a,size,temp1);

delete1(a,size-1,temp2);

system("pause");

return 0;

}

int Binary\_ser(int a[],int size,int value)

{

int lb=0,ub=size-1,mid,position=0;

bool found=false;

while(!found && lb<=ub)

{

mid=(lb+ub)/2;

if(a[mid]==value)

{

found=true;

position=mid;

}

else if (a[mid]>value)

ub=mid-1;

else

lb=mid+1;

}

return position;

}

void delete1(int a[],int size,int loc)

{

for(int j=loc;j<size;j++)

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

cout<<"\n\nArray with Deleted value: "<<endl;

for(int i=0;i<size-1;i++)

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

cout<<"\n\n";

}

**SCREENSHOT:**

