****

**KRISHNA ENGINEERING COLLEGE**

**Data Structure Lab**

**KCS – 301**

SUBMITTED BY:-

**Sourabh Bhardwaj**

**1901610100207**

SUBMITTED TO:-

**Ms. RASHMI SHARMA**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Krishna Engineering College**  **Department of Computer Science and Engineering** | | | | |  |
| **Name: Sourabh Bhardwaj**  **Roll No.:1901610100207** | | | |  |
|  |
| **Year/Semester:- 2nd/III** | | | **Session:- 2020-21** |  |
| **Subject Name:- Data Structures** | | | **Subject Code:- KCS-351** |  |
| **Faculty :-** | | **Ms. RASHMI SHARMA** | **Section/ Group: 2/B** |  |
|  | |  |  |  |
| **S.No** | | **Experiment /Program Name** | **Scheduled Date** | **Sign** |
|
| 1 | | Write a program to take 5 values and store | 08/08/2020 |  |
| 2 | | Read N number of values in an array and display in it in reverse order. | 13/08/2020 |  |
| 3 | | Average of N number | 20/08/2020 |  |
| **4** | | Find sum of all the element in array | 27/08/2020 |  |
|
| 5 | | Copy the elements of one array to another array | 27/08/2020 |  |
| 6 | | Insert an element in an array | 03/09/2020 |  |
| 7 | | Perform Traversal in an array | 03/09/2020 |  |
| 8 | Perform binary search | | 10/09/2020 |  |
| 9 | Perform linear search in an array | | 10/09/2020 |  |
| 10 | Perform Insertion sort in array | | 17/09/2020 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 11 | | Perform selection sort | 17/09/2020 |  |
| 12 | | Perform bubble sort | 24/09/2020 |  |
| 13 | | Implementation of Merge sort | 01/10/2020 |  |
| 14 | Implementation of Quick sort | | 01/10/2020 |  |

PROGRAM**:** ​***1***

WRITE A PROGRAM TO TAKE 5 VALUES FROM THE USER AND STORE.

**INPUT:**

#include​ ​<stdio.h>

int​ main​()​

{

​int​ ​values​[​5​];

​printf​(​"Enter 5 integers: "​);

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

{

​scanf​(​"%d"​, &​values​[​i​]);

}

​printf​(​"Displaying integers: "​);

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

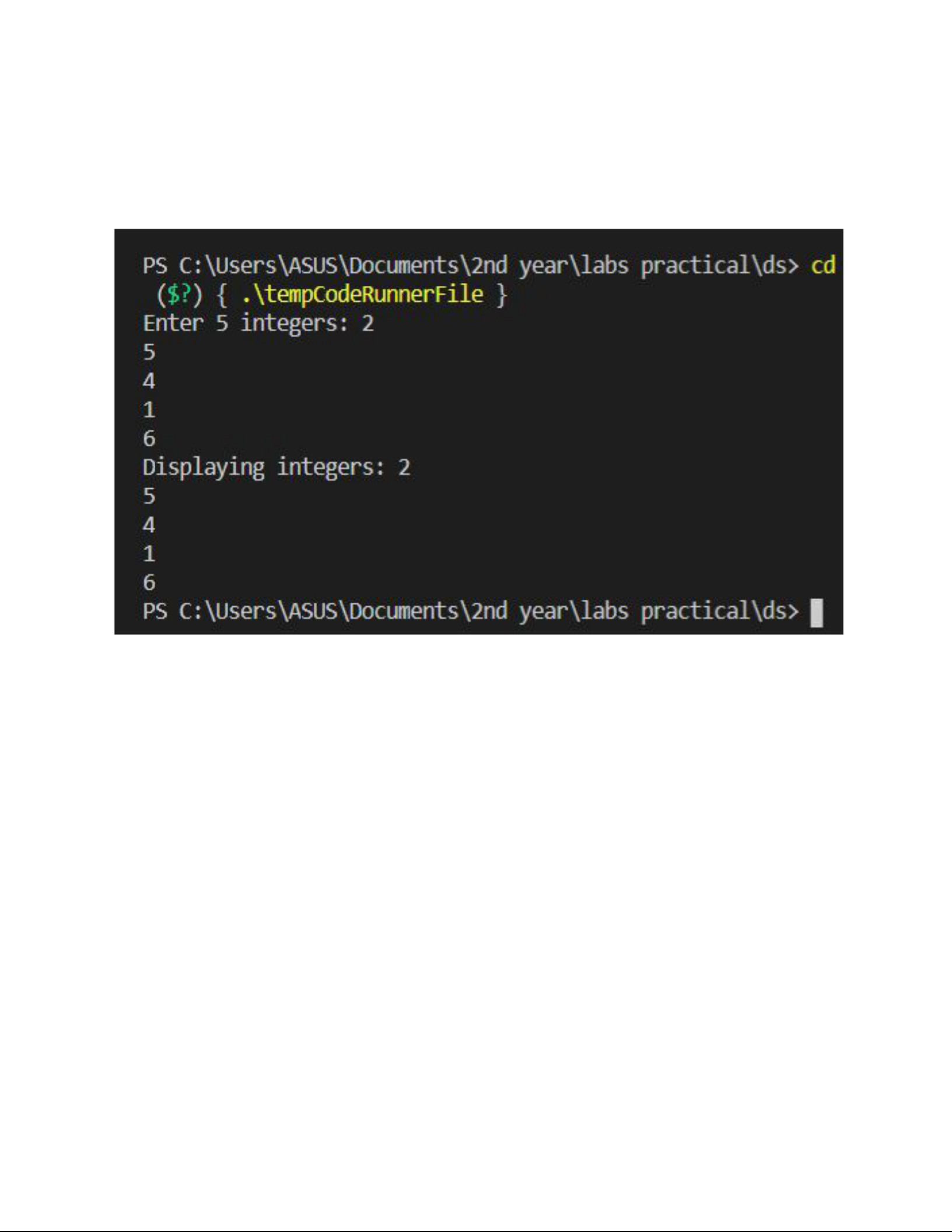
{

​printf​(​"%d​\n​"​, ​values​[​i​]);

}

​return​ ​0​;

}

**OUTPUT :**

PROGRAM: ​*2*

WRITE*​* A PROGRAM IN C TO READ N NUMBER OF VALUES IN AN ARRAY AND

DISPLAY IT IN REVERSE ORDER.​

**INPUT:**

#include​ ​<stdio.h>

void​ main​()​

{

​int​ ​i​, ​n​, ​a​[​100​];

​printf​(​"​\n\n​Read n number of values in an array and display it in reverseorde r:​\n​"​);

​printf​(​"---------------------------------------------------------------------

---​\n​"​);

​printf​(​"Input the number of elements to store in the array :"​); ​scanf​(​"%d"​, &​n​);

​printf​(​"Input %d number of elements in the array :​\n​"​, ​n​); ​for​ (​i​ = ​0​; ​i​ < ​n​; ​i​++)

{

​printf​(​"element - %d : "​, ​i​);

scanf​(​"%d"​,​ &a​[​i​]);​

}

​printf​(​"​\n​The values store into the array are : ​\n​"​); ​for​ (​i​ = ​0​; ​i​ < ​n​; ​i​++)

{

​printf​(​"% 5d"​, ​a​[​i​]);

}

​printf​(​"​\n\n​The values store into the array in reverse are :​\n​"​); ​for​ (​i​ = ​n​ - ​1​; ​i​ >= ​0​; ​i​--)

{

​printf​(​"% 5d"​, ​a​[​i​]);

}

​printf​(​"​\n\n​"​);

}

**OUTPUT :**

PROGRAM: ​*3*

*​*WRITE A PROGRAM IN C TO FIND AVERAGE OF n NUMBERS.

**INPUT:**

#include​ ​<stdio.h>

int​ main​()​

{

​int​ ​marks​[​10​], ​i​, ​n​, ​sum​ = ​0​, ​average​;

printf​(​"Enter​ number of elements: ");​

scanf​(​"%d"​,​ &n​);​

​for​ (​i​ = ​0​; ​i​ < ​n​; ++​i​)

{

​printf​(​"Enter number%d: "​, ​i​ + ​1​);

scanf​(​"%d"​,​ &marks​[​i​]);​

​sum​ += ​marks​[​i​];

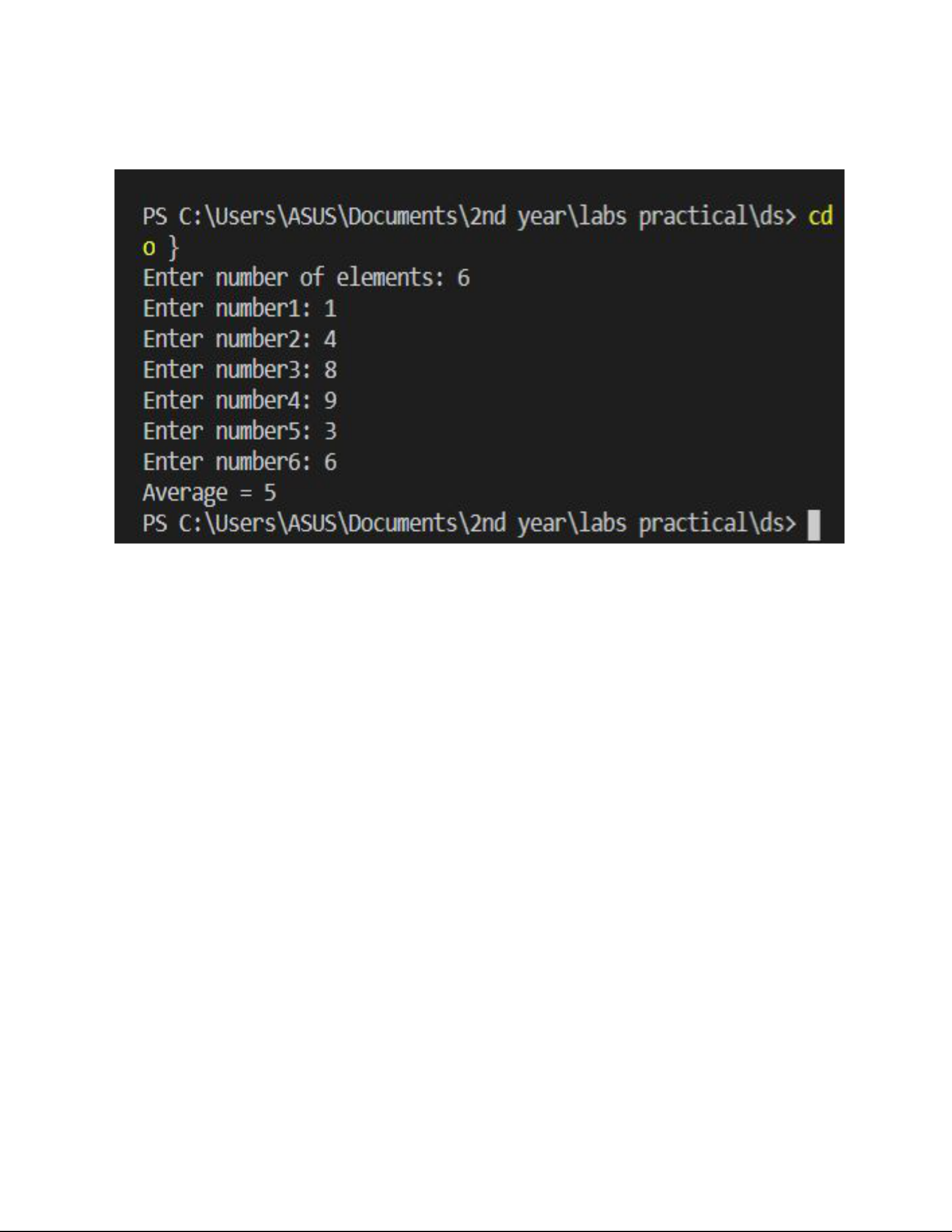
}

​average​ = ​sum​ / ​n​;

​printf​(​"Average = %d"​, ​average​);

return​​ 0​;​

}

**OUTPUT :**

PROGRAM: ​4

*​*WRITE A PROGRAM IN C TO FIND SUM OF ALL THE ELEMENTS IN AN ARRAY

**INPUT:**

#include​ ​<stdio.h>

void​ main​()​

{

​int​ ​a​[​100​];

​int​ ​i​, ​n​, ​sum​ = ​0​;

​printf​(​"​\n\n​Find sum of all elements of array:​\n​"​); ​printf​(​"--------------------------------------​\n​"​);

​printf​(​"Input the number of elements to be stored in the array :"​); ​scanf​(​"%d"​, &​n​);

​printf​(​"Input %d elements in the array :​\n​"​, ​n​); ​for​ (​i​ = ​0​; ​i​ < ​n​; ​i​++)

{

​printf​(​"element - %d : "​, ​i​);

scanf​(​"%d"​,​ &a​[​i​]);​

}

​for​ (​i​ = ​0​; ​i​ < ​n​; ​i​++)

{

​sum​ += ​a​[​i​];

}

​printf​(​"Sum of all elements stored in the array is : %d​\n\n​"​, ​sum​);

}



**OUTPUT :**

PROGRAM: ​*5*

​Write A PROGRAM IN C TO COPY THE ELEMENTS OF ONE ARRAY INTO THE OTHER ARRAY.

**INPUT:**

#include​ ​<stdio.h>

void​ main​()​

{

​int​ ​arr1​[​100​], ​arr2​[​100​];

int​​ i​,​ n​;​

​printf​(​"​\n\n​Copy the elements one array into another array :​\n​"​); ​printf​(​"----------------------------------------------------​\n​"​);

​printf​(​"Input the number of elements to be stored in the array :"​); ​scanf​(​"%d"​, &​n​);

​printf​(​"Input %d elements in the array :​\n​"​, ​n​); ​for​ (​i​ = ​0​; ​i​ < ​n​; ​i​++)

{

​printf​(​"element - %d : "​, ​i​);

scanf​(​"%d"​,​ &arr1​[​i​]);​

}

​for​ (​i​ = ​0​; ​i​ < ​n​; ​i​++)

{

​arr2​[​i​] = ​arr1​[​i​];

}

​printf​(​"​\n​The elements stored in the first array are :​\n​"​); ​for​ (​i​ = ​0​; ​i​ < ​n​; ​i​++)

{

​printf​(​"% 5d"​, ​arr1​[​i​]);

}

​printf​(​"​\n\n​The elements copied into the second array are :​\n​"​); ​for​ (​i​ = ​0​; ​i​ < ​n​; ​i​++)

{

​printf​(​"% 5d"​, ​arr2​[​i​]);

}

​printf​(​"​\n\n​"​);

}

**OUTPUT :**

PROGRAM: ​*6*

*​*WRITE A PROGRAM TO INSERT AN ELEMENT IN ARRAY.

**INPUT:**

#include​ ​<stdio.h>

int​ main​()​

{

​int​ ​arr​[​100​] = {​0​};

​int​ ​i​, ​x​, ​pos​, ​n​ = ​10​;

​for​ (​i​ = ​0​; ​i​ < ​10​; ​i​++)

arr​[​i​]​ = i​​ + 1​;​

​for​ (​i​ = ​0​; ​i​ < ​n​; ​i​++)

printf​(​"%d​ ",​ arr​[​i​]);​

​printf​(​"​\n​"​);

​x​ = ​50​;

pos​​ = 5​;​

n​++;​

​for​ (​i​ = ​n​; ​i​ >= ​pos​; ​i​--)

arr​[​i​]​ = arr​[​i​​ - 1​];​

​arr​[​pos​ - ​1​] = ​x​;

​for​ (​i​ = ​0​; ​i​ < ​n​; ​i​++)

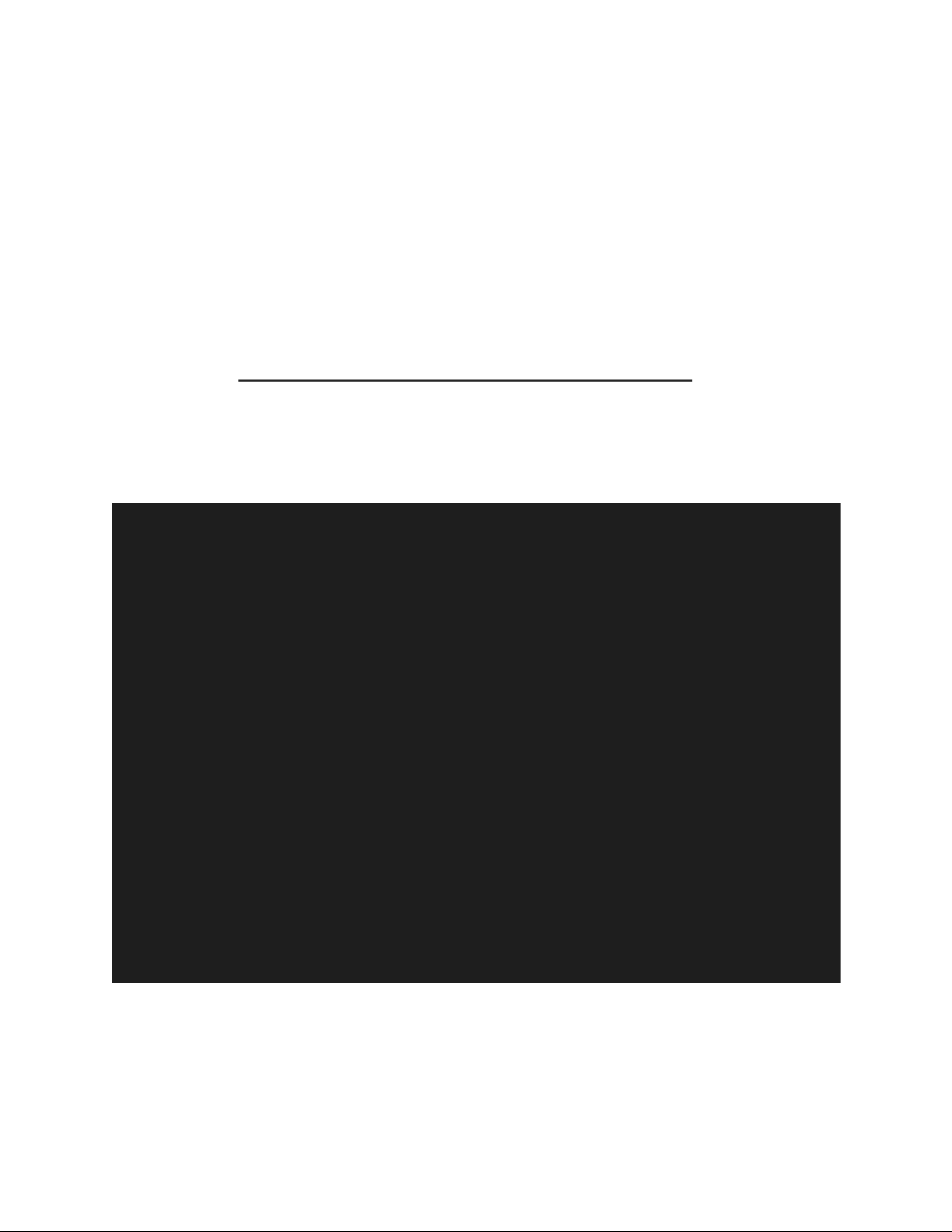
printf​(​"%d​ ",​ arr​[​i​]);​

​printf​(​"​\n​"​);

return​​ 0​;​

}

**OUTPUT :**



PROGRAM: ​*7*

IMPLEMENTATION*​* OF TRAVERSE

**INPUT:**

#include​ ​<stdio.h>

int​ ​main​()

{

​int​ ​i​, ​size​, ​a​[​20​];

​printf​(​"Enter no. of elements of array : ​\n​"​);

scanf​(​"%d"​,​ &size​);​

​if​ (​size​ > ​20​)

{

​printf​(​"Overflow"​);

}

​else

{

​printf​(​"Elements of array : ​\n​"​);

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

{

​scanf​(​"%d"​, &​a​[​i​]);

}

​printf​(​"Elements are: "​);

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

{

​printf​(​"%d "​, ​a​[​i​]);

}

}

}



**OUTPUT :**

PROGRAM : ​8

**IMPLEMENTATION*​* OF BINARY SEARCH**

**INPUT :**

#include​ ​<stdio.h>

int​ ​main​()

{

​int​ ​arr​[​50​], ​i​, ​n​, ​x​, ​flag​ = ​0​, ​first​, ​last​, ​mid​;

​printf​(​"Enter size of array:"​);

scanf​(​"%d"​,​ &n​);​

​printf​(​"​\n​Enter array element(ascending order)​\n​"​);

​for​ (​i​ = ​0​; ​i​ < ​n​; ++​i​)

scanf​(​"%d"​,​ &arr​[​i​]);​

​printf​(​"​\n​Enter the element to search:"​);

scanf​(​"%d"​,​ &x​);​

​first​ = ​0​;

​last​ = ​n​ - ​1​;

​while​ (​first​ <= ​last​)

{

​mid​ = (​first​ + ​last​) / ​2​;

​if​ (​x​ == ​arr​[​mid​])

{

​flag​ = ​1​;

break​;​

}

​else​ ​if​ (​x​ > ​arr​[​mid​])

first​​ = mid​​ + 1​;​

​else

​last​ = ​mid​ - ​1​;

}

​if​ (​flag​ == ​1​)

​printf​(​"​\n​Element found at position %d"​, ​mid​ + ​1​);

​else

​printf​(​"​\n​Element not found"​);

​return​ ​0​;

}



**OUTPUT :**



PROGRAM : ​*9*

**​**IMPLEMENTATION OF LINEAR SEARCH

**INPUT :**

#include​ ​<stdio.h>

int​ ​main​()

{

​int​ ​a​[​20​], i, x, n;

​printf​(​"How many elements? ​\n​"​);

scanf​(​"%d"​,​ &n);

​printf​(​"Enter array elements:​\n​ "​);

for​​ (i = 0​;​ i < n; ++i)

​scanf​(​"%d"​, &​a​[i]);

​printf​(​"​\n​Enter element to search: ​\n​"​);

scanf​(​"%d"​,​ &x);

​for​ (i = ​0​; i < n; ++i)

if​​ (a​[i]​ == x)

​break​;

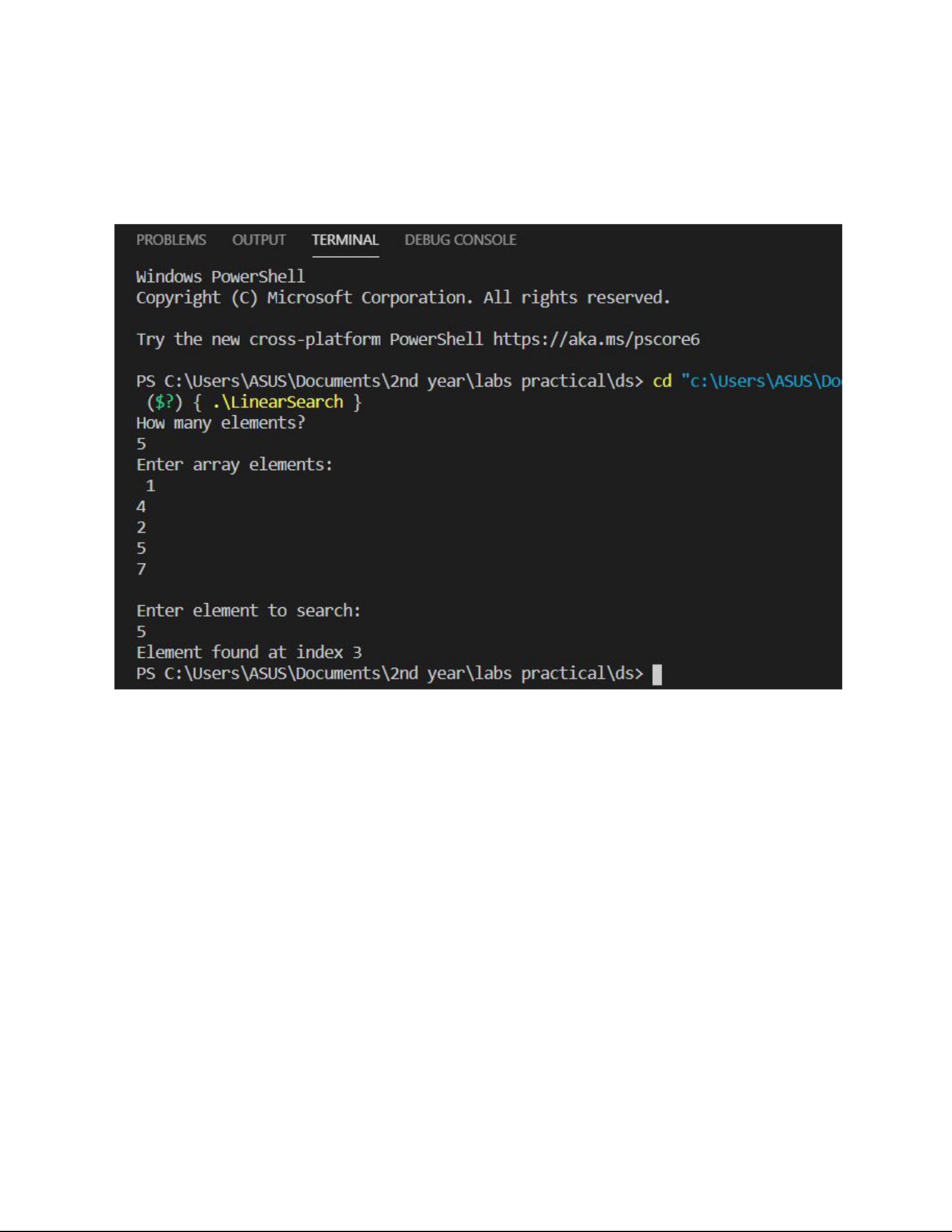
​if​ (i < n)

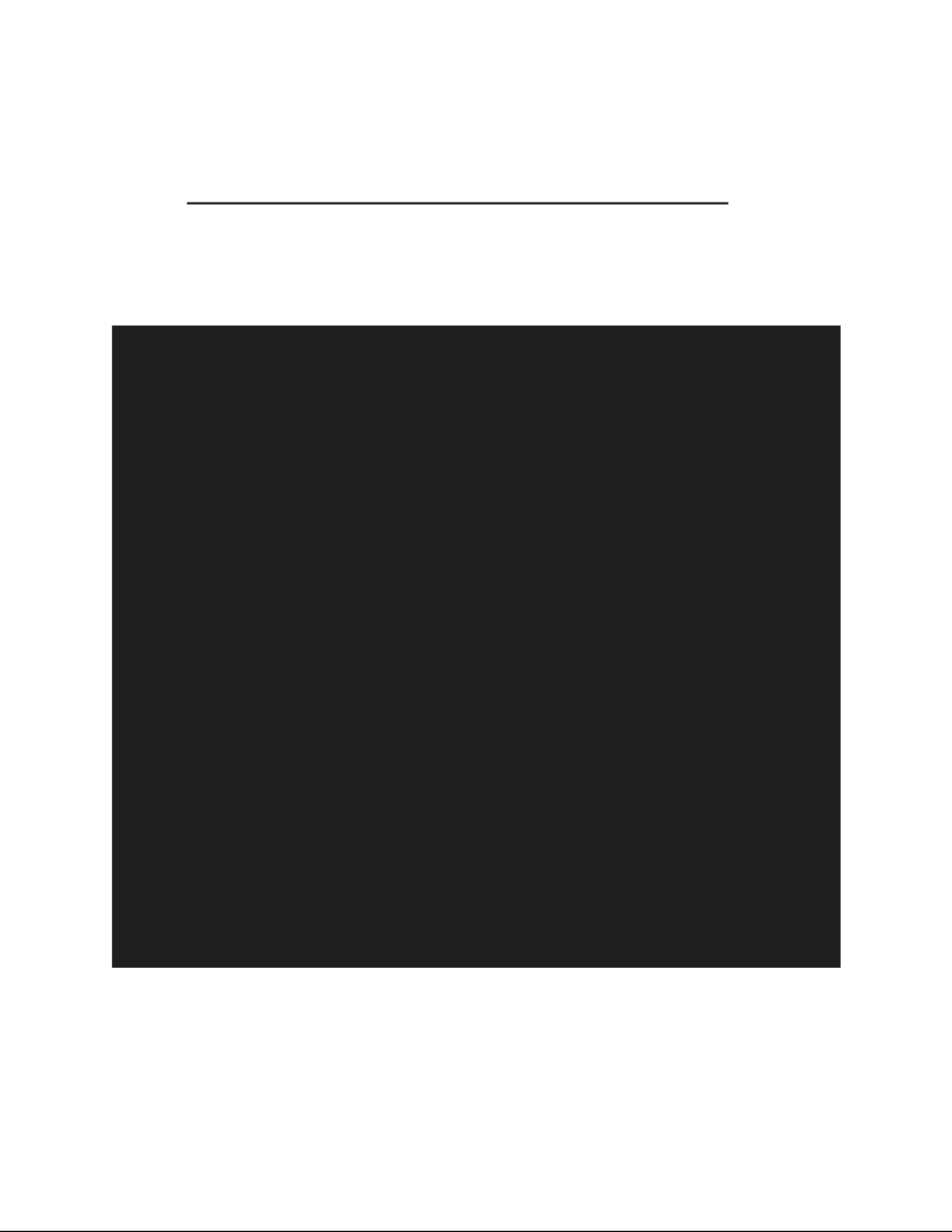
​printf​(​"Element found at index %d"​, i);

​else

​printf​(​"Element not found"​);

}

**OUTPUT :**

PROGRAM : ​*10*

IMPLEMENTATION​ OF INSERTION SORT

**INPUT :**

#include​ ​<stdio.h>

int​ ​main​()

{

​int​ ​i​, ​j​, ​n​, ​temp​, ​a​[​30​];

​printf​(​"Enter the number of elements:"​);

scanf​(​"%d"​,​ &n​);​

​printf​(​"​\n​Enter the elements​\n​"​);

​for​ (​i​ = ​0​; ​i​ < ​n​; ​i​++)

{

​scanf​(​"%d"​, &​a​[​i​]);

}

​for​ (​i​ = ​1​; ​i​ <= ​n​ - ​1​; ​i​++)

{

​temp​ = ​a​[​i​];

​j​ = ​i​ - ​1​;

​while​ ((​temp​ < ​a​[​j​]) && (​j​ >= ​0​))

{

​a​[​j​ + ​1​] = ​a​[​j​]; ​//moves element forward

​j​ = ​j​ - ​1​;

}

​a​[​j​ + ​1​] = ​temp​; ​//insert element in proper place

}

​printf​(​"​\n​Sorted list is as follows​\n​"​);

​for​ (​i​ = ​0​; ​i​ < ​n​; ​i​++)

{

​printf​(​"%d "​, ​a​[​i​]);

}

​return​ ​0​;

}

**OUTPUT :**

PROGRAM : ​*11*

*​*IMPLEMENTATION OF SELECTION SORT

**INPUT :**

#include​ ​<stdio.h>

void​ ​selection​(​int​ ​a​[]​, ​int​ ​size​);

int​ main​()​

{

​int​ ​a​[​10​], ​size​, ​i​;

​printf​(​"​\n​enter size of array:"​);

scanf​(​"%d"​,​ &size​);​

printf​(​"enter​ array elements:");​

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

​scanf​(​"%d"​, &​a​[​i​]);

selection​(​a​,​ size​);​

return​​ 0​;​

}

void​ ​selection​(​int​ ​a​[]​, ​int​ ​size​)

{

​int​ ​min​, ​temp​, ​i​;

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

{

​min​ = ​i​;

​for​ (​int​ ​j​ = ​i​ + ​1​; ​j​ < ​size​; ​j​++)

{

​if​ (​a​[​min​] > ​a​[​j​])

min​​ = j​;​

}

​if​ (​min​ != ​i​)

{

​temp​ = ​a​[​min​];

a​[​min​]​ = a​[​i​];​

a​[​i​]​ = temp​;​

}

}

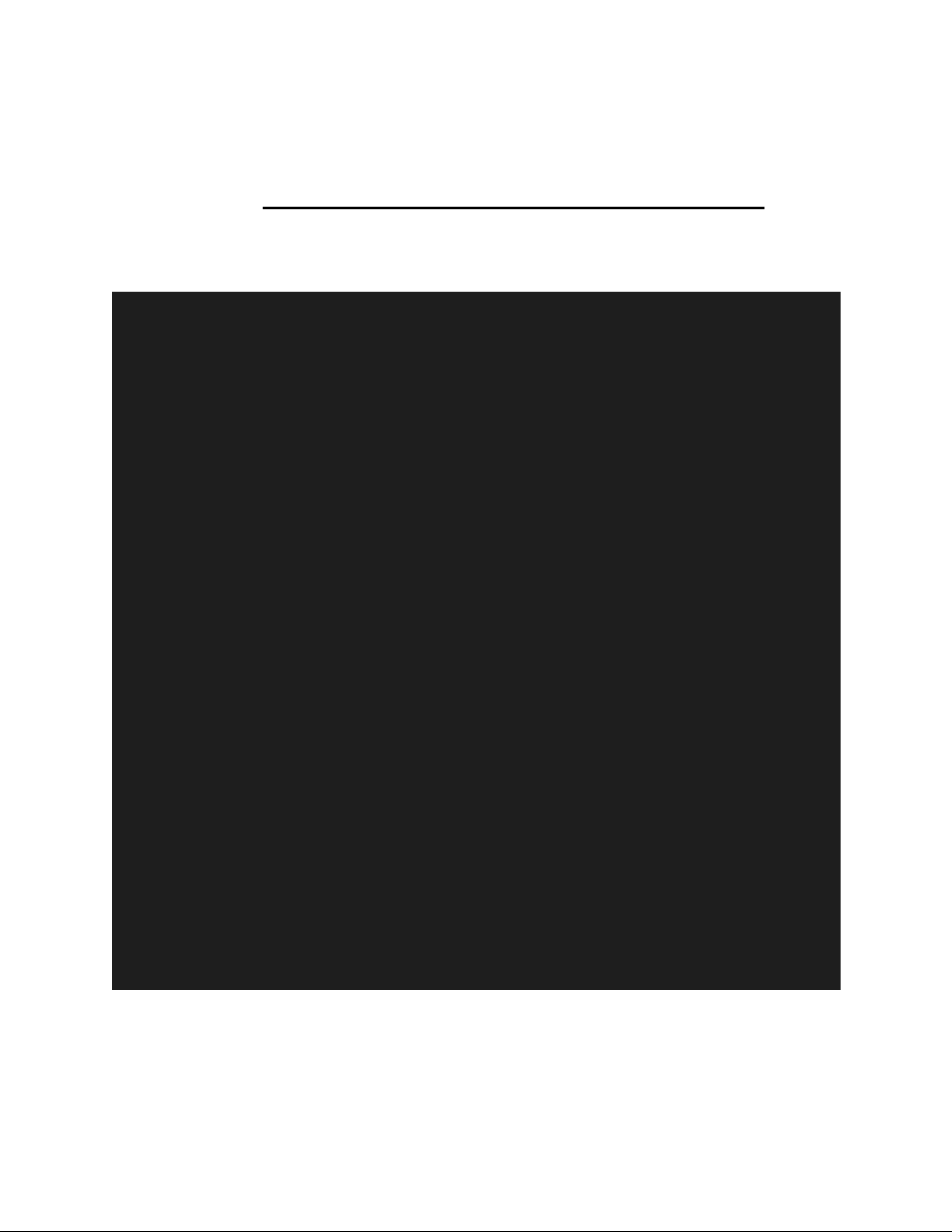
​printf​(​"​\n​sorted array is:"​);

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

​printf​(​"%d "​, ​a​[​i​]);

}

**OUTPUT :**

PROGRAM :​12

IMPLEMENTATION*​* OF BUBBLE SORT

**INPUT :**

#include​ ​<stdio.h>

int​ ​main​()

{

​int​ ​array​[​100​], ​n​, ​c​, ​d​, ​swap​;

​printf​(​"Enter number of elements​\n​"​);

scanf​(​"%d"​,​ &n​);​

​printf​(​"Enter %d integers​\n​"​, ​n​);

​for​ (​c​ = ​0​; ​c​ < ​n​; ​c​++)

scanf​(​"%d"​,​ &array​[​c​]);​

​for​ (​c​ = ​0​; ​c​ < ​n​ - ​1​; ​c​++)

{

​for​ (​d​ = ​0​; ​d​ < ​n​ - ​c​ - ​1​; ​d​++)

{

​if​ (​array​[​d​] > ​array​[​d​ + ​1​]) ​/\* For decreasing order use '<' instead of '>' \*

/

{

​swap​ = ​array​[​d​];

​array​[​d​] = ​array​[​d​ + ​1​];

array​[​d​​ + 1​]​ = swap​;​

}

}

}

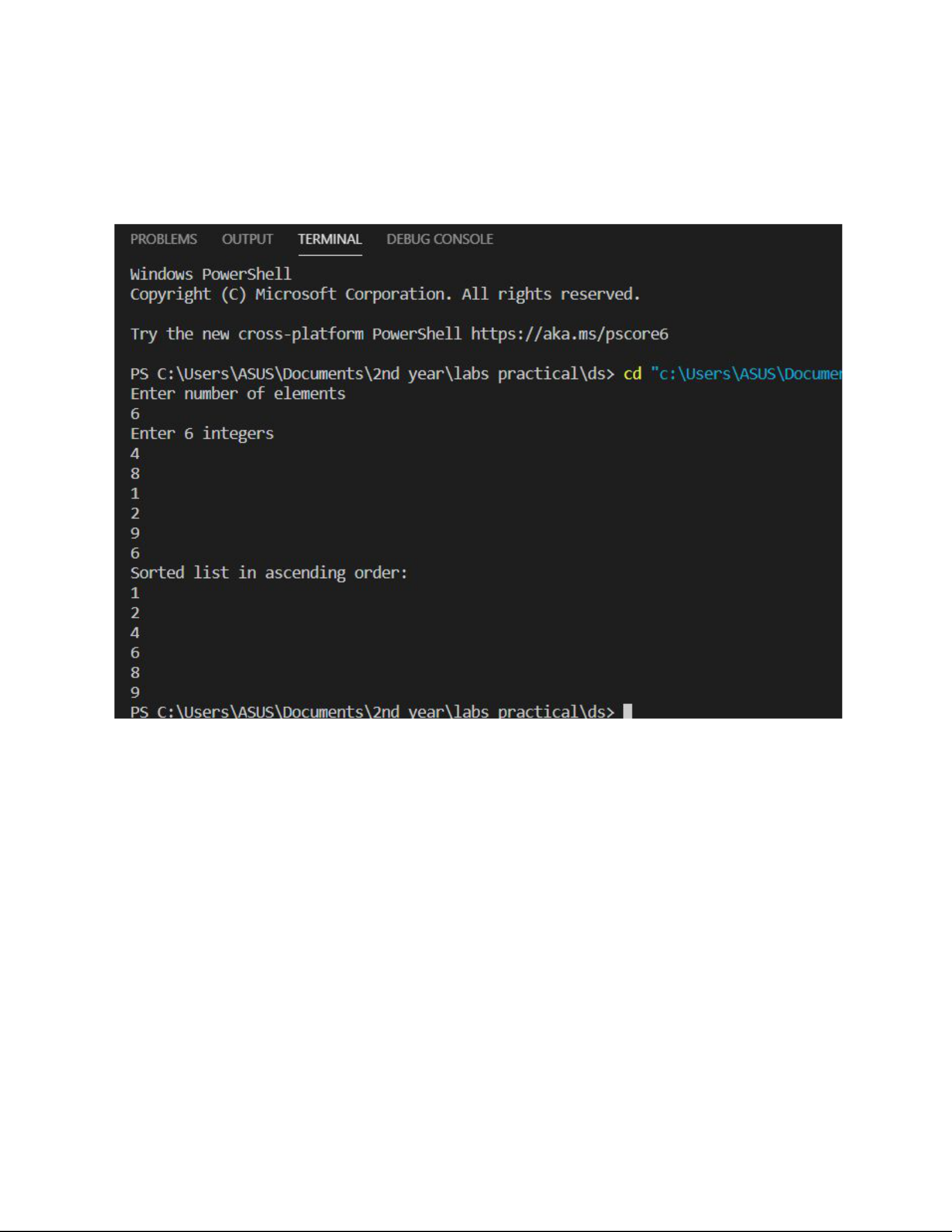
​printf​(​"Sorted list in ascending order:​\n​"​);

​for​ (​c​ = ​0​; ​c​ < ​n​; ​c​++)

printf​(​"%d​\n​"​,​ array​[​c​]);​

​return​ ​0​;

}

**OUTPUT :**

PROGRAM :​***13***

​IMPLEMENTATION OF MERGE SORT

**INPUT :**

#include​ ​<stdio.h>

void​ ​merge​(​int​ ​arr​[]​, ​int​ ​p​, ​int​ ​q​, ​int​ ​r​)

{

​int​ ​n1​ = ​q​ - ​p​ + ​1​;

​int​ ​n2​ = ​r​ - ​q​;

​int​ ​L​[​n1​], ​M​[​n2​];

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

​L​[​i​] = ​arr​[​p​ + ​i​];

​for​ (​int​ ​j​ = ​0​; ​j​ < ​n2​; ​j​++)

​M​[​j​] = ​arr​[​q​ + ​1​ + ​j​];

​int​ ​i​, ​j​, ​k​;

​i​ = ​0​;

​j​ = ​0​;

​k​ = ​p​;

​while​ (​i​ < ​n1​ && ​j​ < ​n2​)

{

​if​ (​L​[​i​] <= ​M​[​j​])

{

​arr​[​k​] = ​L​[​i​];

i​++;​

}

​else

{

​arr​[​k​] = ​M​[​j​];

j​++;​

}

​k​++;

}

​while​ (​i​ < ​n1​)

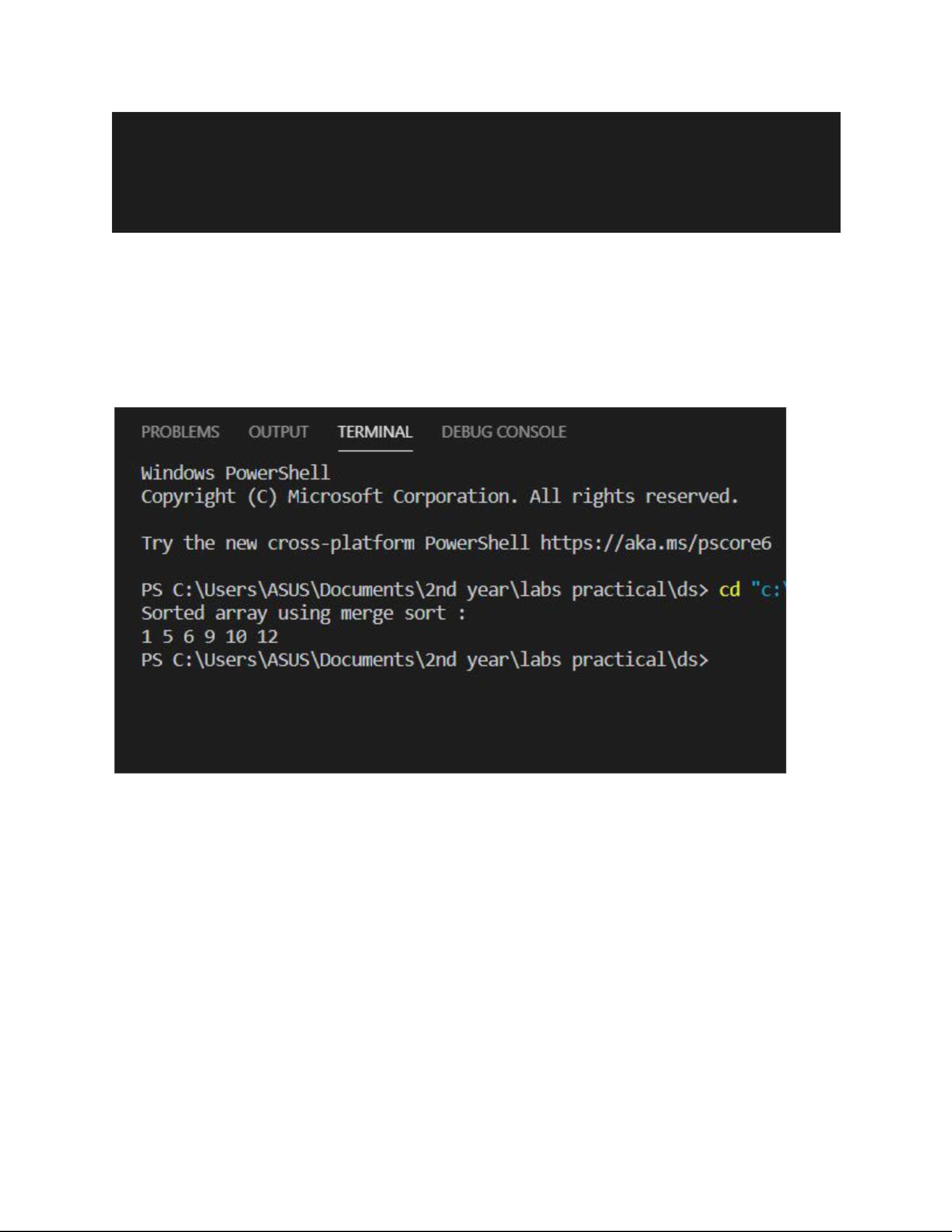
{

​arr​[​k​] = ​L​[​i​];

i​++;​

​k​++;

}



​while​ (​j​ < ​n2​)

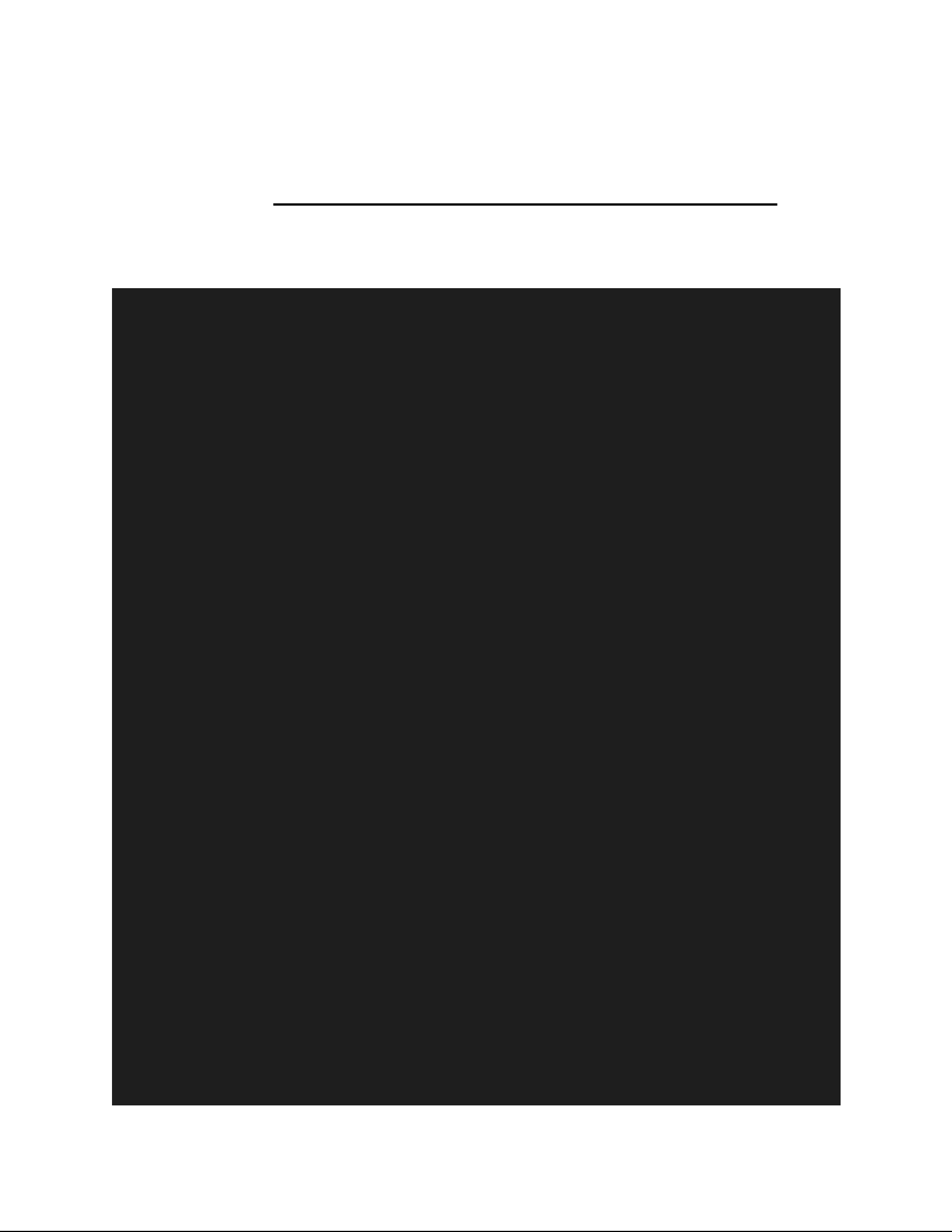
{

​arr​[​k​] = ​M​[​j​];

j​++;​

​k​++;

**OUTPUT :**

PROGRAM : ​*14*

IMPLEMENTATION​ OF QUICK SORT

**INPUT :**

#include​ ​<stdio.h>

void​ ​swap​(​int​ \*​a​, ​int​ \*​b​)

{

​int​ ​t​ = \*​a​;

\*​a​ = \*​b​;

\*​b​ = ​t​;

}

int​ ​partition​(​int​ ​array​[]​, ​int​ ​low​, ​int​ ​high​)

{

​int​ ​pivot​ = ​array​[​high​];

int​​ i​​ = (low​​ - 1​);​

​for​ (​int​ ​j​ = ​low​; ​j​ < ​high​; ​j​++)

{

​if​ (​array​[​j​] <= ​pivot​)

{

​i​++;

​swap​(&​array​[​i​], &​array​[​j​]);

}

}

​swap​(&​array​[​i​ + ​1​], &​array​[​high​]);

return​​ (i​​ + 1​);​

}

void​ ​quickSort​(​int​ ​array​[]​, ​int​ ​low​, ​int​ ​high​)

{

​if​ (​low​ < ​high​)

{

​int​ ​pi​ = ​partition​(​array​, ​low​, ​high​);

​quickSort​(​array​, ​low​, ​pi​ - ​1​);

​quickSort​(​array​, ​pi​ + ​1​, ​high​);

}

}

void​ ​printArray​(​int​ ​array​[]​, ​int​ ​size​)

{

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

{

​printf​(​"%d "​, ​array​[​i​]);

}

**OUTPUT :**