



Chandpur Science and Technology University

Department of Computer Science and Engineering

LAB ASSIGNMENT #4



C Lab Assignment Submitted By:

Name: Iftekhhar Hossain

Student ID: B220101024

Submitted To:

Prince Mahmud

Lecturer

Department of Computer Science and Engineering, CSTU

Lab Date:

Submission Date: 15/12/2023

Marks & Signature

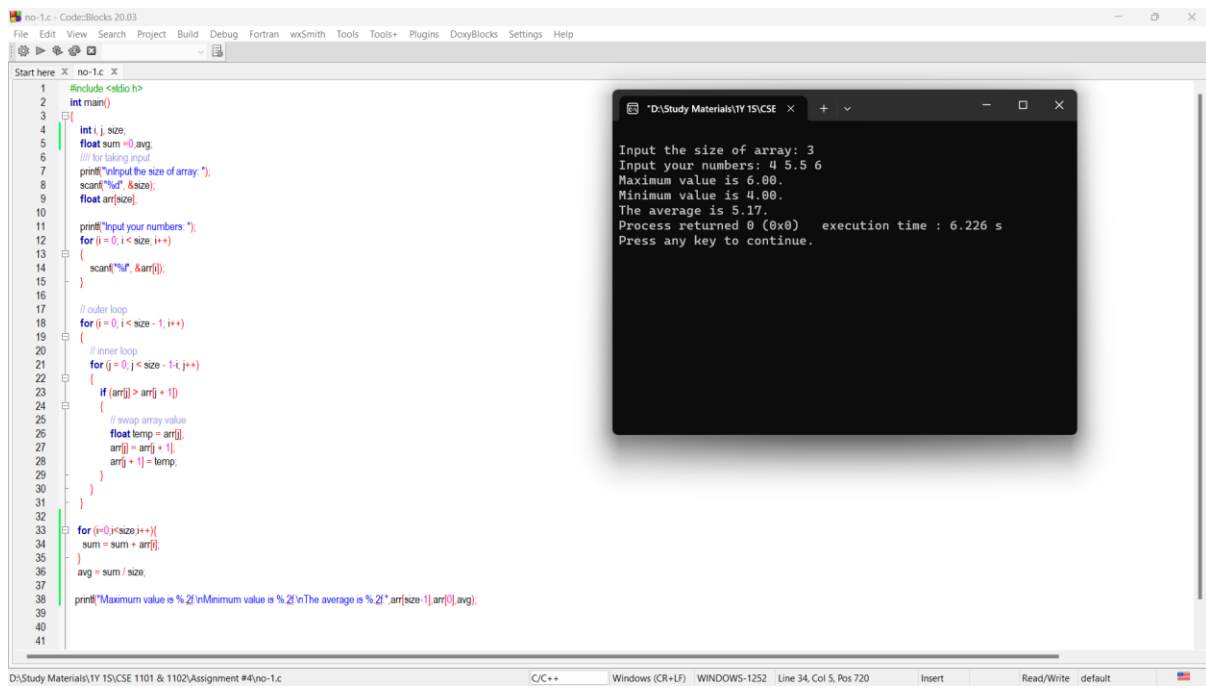
--

Problem No – 1

Title:

Write a program to find the maximum, minimum and average from a list of floating point numbers.

Code and Output:



The screenshot displays a C++ IDE with a source code editor on the left and a terminal window on the right. The code in the editor is a program to find the maximum, minimum, and average of an array of floating-point numbers. It includes a bubble sort algorithm. The terminal window shows the program's execution with the following input and output:

```
no-1.c - CodeBlocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoryBlocks Settings Help
Start here x no-1.c x
1 #include <iostream>
2 int main()
3 {
4     int i, size;
5     float sum = 0, avg;
6     // for taking input
7     printf("Input the size of array: ");
8     scanf("%d", &size);
9     float arr[size];
10
11     printf("Input your numbers: ");
12     for (i = 0; i < size; i++)
13     {
14         scanf("%f", &arr[i]);
15     }
16
17     // outer loop
18     for (i = 0; i < size - 1; i++)
19     {
20         // inner loop
21         for (j = 0; j < size - i - 1; j++)
22         {
23             if (arr[j] > arr[j + 1])
24             {
25                 // swap array value
26                 float temp = arr[j];
27                 arr[j] = arr[j + 1];
28                 arr[j + 1] = temp;
29             }
30         }
31     }
32
33     for (i = 0; i < size; i++)
34     {
35         sum = sum + arr[i];
36     }
37     avg = sum / size;
38     printf("Maximum value is %2f\nMinimum value is %2f\nThe average is %2f", arr[size-1], arr[0], avg);
39
40
41
D:\Study Materials\TY 15\CSE 1101 & 1102\Assignment #4\no-1.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 34, Col 5, Pos 720 Insert Read/Write default
```

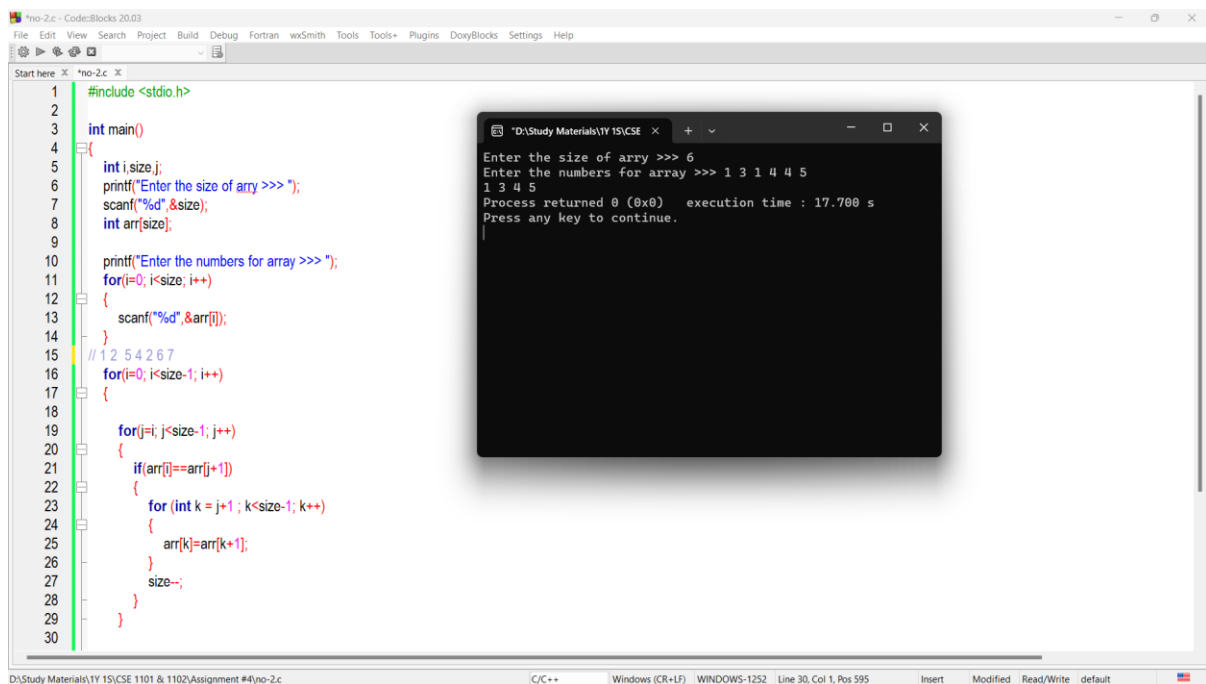
```
Input the size of array: 3
Input your numbers: 4 5.5 6
Maximum value is 6.00
Minimum value is 4.00
The average is 5.17
Process returned 0 (0x0)   execution time : 6.226 s
Press any key to continue.
```

Problem No – 2

Title:

Write a program to delete duplicate elements from an array.

Code and Output:



```
#include <stdio.h>

int main()
{
    int i, size, j;
    printf("Enter the size of array >>> ");
    scanf("%d", &size);
    int arr[size];

    printf("Enter the numbers for array >>> ");
    for(i=0; i<size; i++)
    {
        scanf("%d", &arr[i]);
    }
    // 1 2 5 4 2 6 7
    for(i=0; i<size-1; i++)
    {
        for(j=i; j<size-1; j++)
        {
            if(arr[i]==arr[j+1])
            {
                for (int k = j+1 ; k<size-1; k++)
                {
                    arr[k]=arr[k+1];
                }
                size--;
            }
        }
    }
}
```

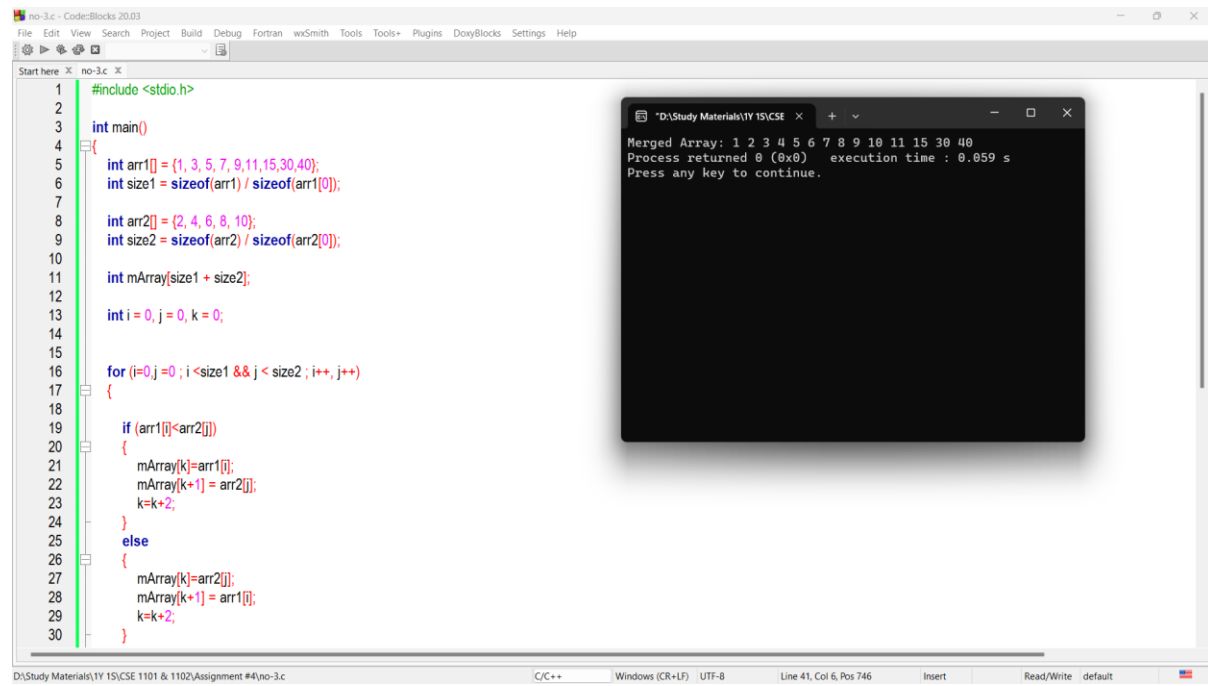
```
Enter the size of array >>> 6
Enter the numbers for array >>> 1 3 1 4 4 5
1 3 4 5
Process returned 0 (0x0)   execution time : 17.700 s
Press any key to continue.
```

Problem No – 3

Title:

Write a program to merge two sorted arrays.

Code and Output:



```
#include <stdio.h>

int main()
{
    int arr1[] = {1, 3, 5, 7, 9, 11, 15, 30, 40};
    int size1 = sizeof(arr1) / sizeof(arr1[0]);

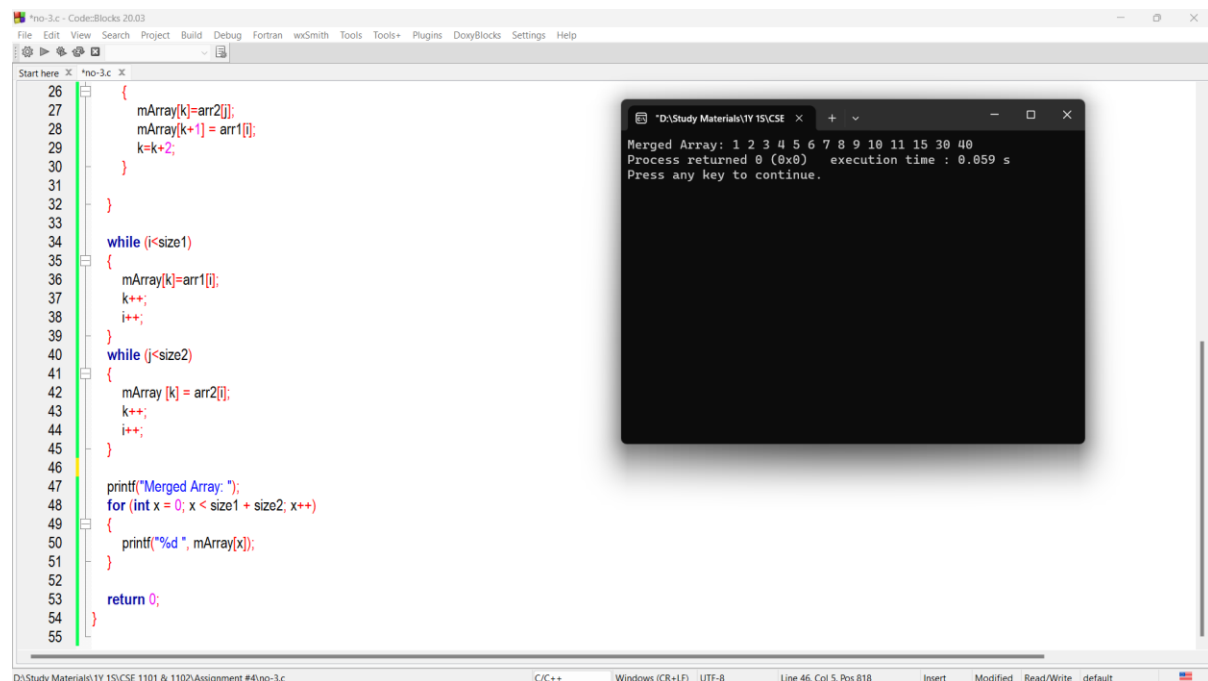
    int arr2[] = {2, 4, 6, 8, 10};
    int size2 = sizeof(arr2) / sizeof(arr2[0]);

    int mArray[size1 + size2];

    int i = 0, j = 0, k = 0;

    for (i=0; j=0; i < size1 && j < size2; i++, j++)
    {
        if (arr1[i] < arr2[j])
        {
            mArray[k] = arr1[i];
            mArray[k+1] = arr2[j];
            k=k+2;
        }
        else
        {
            mArray[k] = arr2[j];
            mArray[k+1] = arr1[i];
            k=k+2;
        }
    }
}
```

Merged Array: 1 2 3 4 5 6 7 8 9 10 11 15 30 40
Process returned 0 (0x0) execution time : 0.059 s
Press any key to continue.



```
#include <stdio.h>

int main()
{
    int arr1[] = {1, 3, 5, 7, 9, 11, 15, 30, 40};
    int size1 = sizeof(arr1) / sizeof(arr1[0]);

    int arr2[] = {2, 4, 6, 8, 10};
    int size2 = sizeof(arr2) / sizeof(arr2[0]);

    int mArray[size1 + size2];

    int i = 0, j = 0, k = 0;

    while (i < size1)
    {
        mArray[k] = arr1[i];
        k++;
        i++;
    }

    while (j < size2)
    {
        mArray[k] = arr2[j];
        k++;
        j++;
    }

    printf("Merged Array: ");
    for (int x = 0; x < size1 + size2; x++)
    {
        printf("%d ", mArray[x]);
    }

    return 0;
}
```

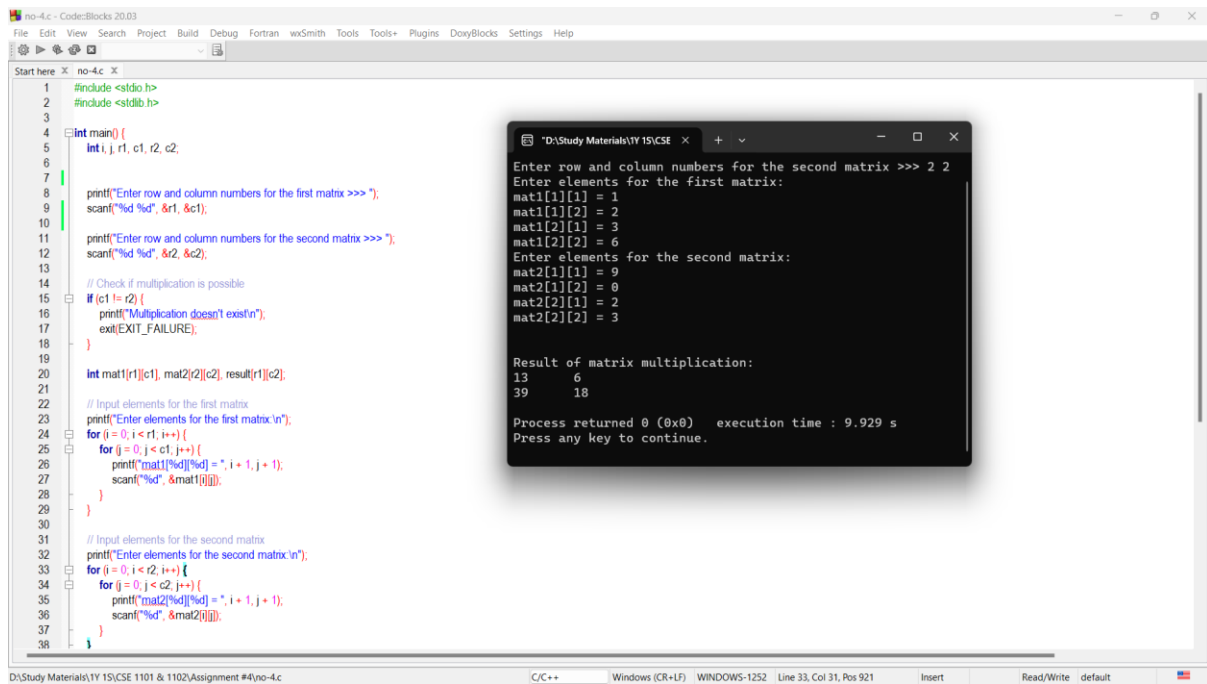
Merged Array: 1 2 3 4 5 6 7 8 9 10 11 15 30 40
Process returned 0 (0x0) execution time : 0.059 s
Press any key to continue.

Problem No – 4

Title:

Write a program to read two matrixes from user into two different 2D array and multiply these two matrixes and finally display the result.

Code and Output:



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main() {
5     int i, j, r1, c1, r2, c2;
6
7     printf("Enter row and column numbers for the first matrix >>> ");
8     scanf("%d %d", &r1, &c1);
9
10    printf("Enter row and column numbers for the second matrix >>> ");
11    scanf("%d %d", &r2, &c2);
12
13    // Check if multiplication is possible
14    if (c1 != r2) {
15        printf("Multiplication doesn't exist\n");
16        exit(EXIT_FAILURE);
17    }
18
19    int mat1[r1][c1], mat2[r2][c2], result[r1][c2];
20
21    // Input elements for the first matrix
22    printf("Enter elements for the first matrix\n");
23    for (i = 0; i < r1; i++) {
24        for (j = 0; j < c1; j++) {
25            printf("mat1[%d][%d] = ", i + 1, j + 1);
26            scanf("%d", &mat1[i][j]);
27        }
28    }
29
30    // Input elements for the second matrix
31    printf("Enter elements for the second matrix\n");
32    for (i = 0; i < r2; i++) {
33        for (j = 0; j < c2; j++) {
34            printf("mat2[%d][%d] = ", i + 1, j + 1);
35            scanf("%d", &mat2[i][j]);
36        }
37    }
38}
```

```
Enter row and column numbers for the second matrix >>> 2 2
Enter elements for the first matrix:
mat1[1][1] = 1
mat1[1][2] = 2
mat1[2][1] = 3
mat1[2][2] = 6
Enter elements for the second matrix:
mat2[1][1] = 9
mat2[1][2] = 0
mat2[2][1] = 2
mat2[2][2] = 3

Result of matrix multiplication:
13    6
39    18

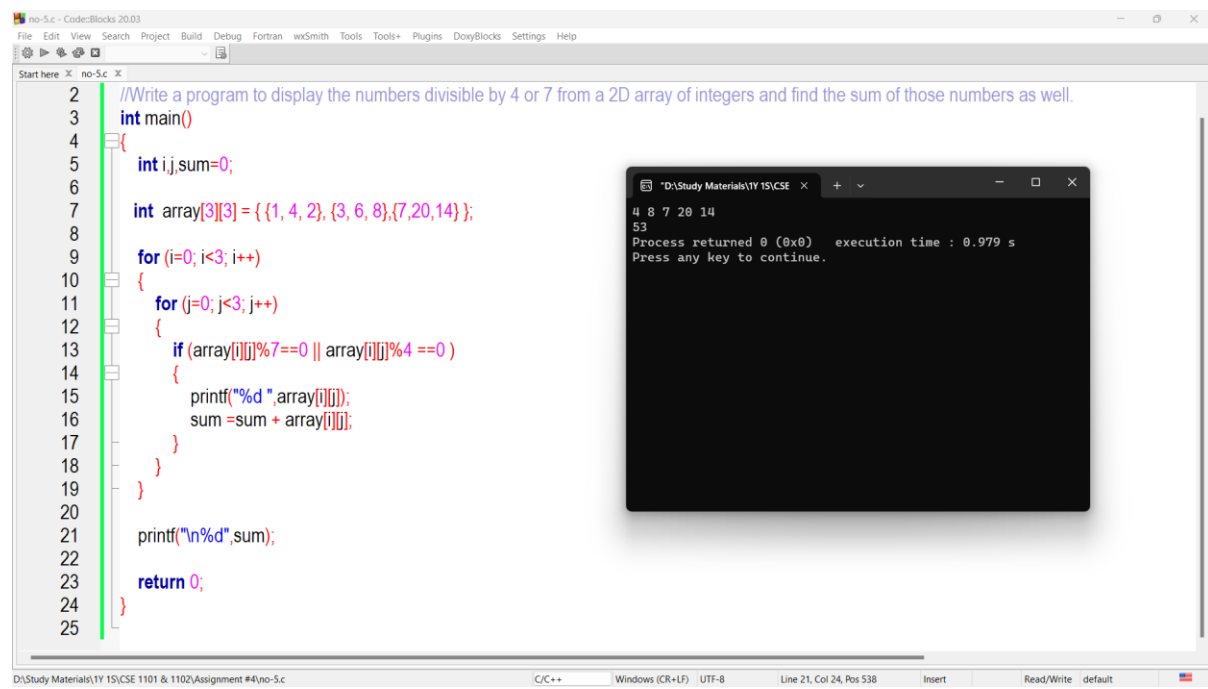
Process returned 0 (0x0)   execution time : 9.929 s
Press any key to continue.
```

Problem No – 5

Title:

Write a program to display the numbers divisible by 4 or 7 from a 2D array of integers and find the sum of those numbers as well.

Code and Output:



The screenshot shows a C++ IDE with a code editor on the left and a terminal window on the right. The code in the editor is as follows:

```
2 //Write a program to display the numbers divisible by 4 or 7 from a 2D array of integers and find the sum of those numbers as well.
3 int main()
4 {
5     int i,j,sum=0;
6
7     int array[3][3] = { {1, 4, 2}, {3, 6, 8}, {7, 20, 14} };
8
9     for (i=0; i<3; i++)
10     {
11         for (j=0; j<3; j++)
12         {
13             if (array[i][j]%7==0 || array[i][j]%4 ==0 )
14             {
15                 printf("%d ",array[i][j]);
16                 sum =sum + array[i][j];
17             }
18         }
19     }
20
21     printf("\n%d",sum);
22
23     return 0;
24 }
25
```

The terminal window on the right displays the output of the program:

```
D:\Study Materials\TY 15\CSE x + v - _ x
4 8 7 20 14
53
Process returned 0 (0x0)   execution time : 0.979 s
Press any key to continue.
```

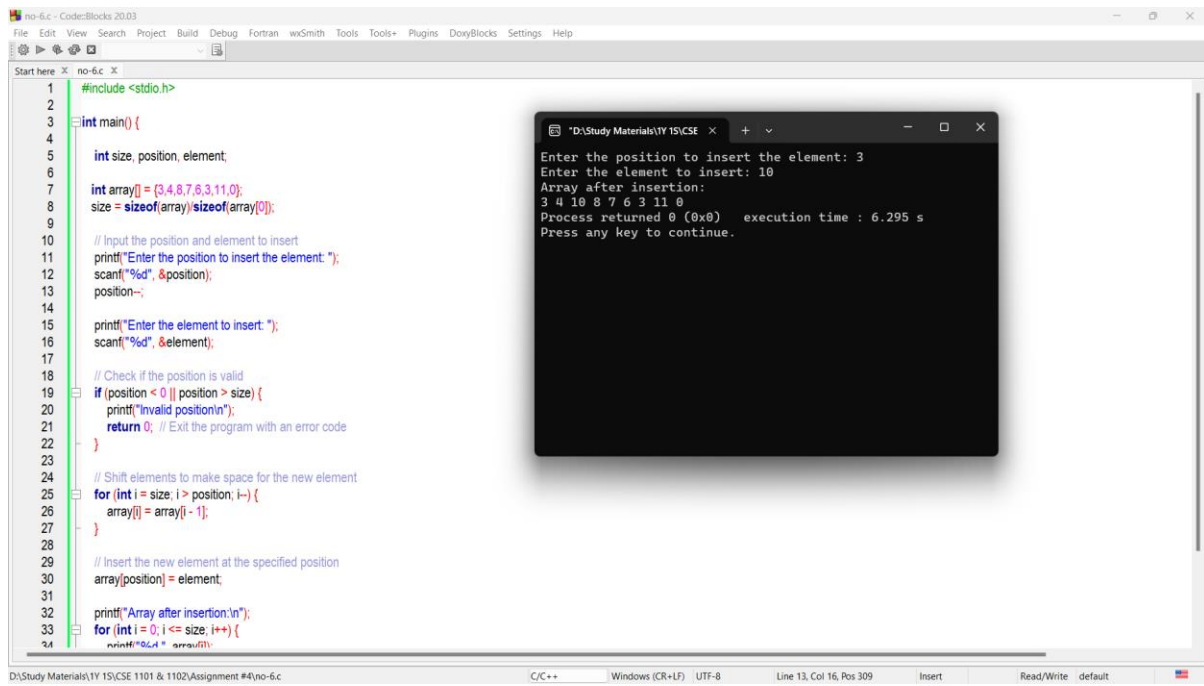
The status bar at the bottom of the IDE indicates the file path as "D:\Study Materials\TY 15\CSE 1101 & 1102\Assignment #4\yno-5.c", the compiler as "C/C++", the encoding as "Windows (CR+LF)", the font as "UTF-8", and the cursor position as "Line 21, Col 24, Pos 538".

Problem No – 6

Title:

Write a program insert an Element Desired or Specific Position in an Array.

Code and Output:



```
1 #include <stdio.h>
2
3 int main() {
4     int size, position, element;
5
6     int array[] = {3,4,8,7,6,3,11,0};
7     size = sizeof(array)/sizeof(array[0]);
8
9     // Input the position and element to insert
10    printf("Enter the position to insert the element: ");
11    scanf("%d", &position);
12    position--;
13
14    printf("Enter the element to insert: ");
15    scanf("%d", &element);
16
17    // Check if the position is valid
18    if (position < 0 || position > size) {
19        printf("Invalid position\n");
20        return 0; // Exit the program with an error code
21    }
22
23    // Shift elements to make space for the new element
24    for (int i = size; i > position; i--) {
25        array[i] = array[i - 1];
26    }
27
28    // Insert the new element at the specified position
29    array[position] = element;
30
31    printf("Array after insertion\n");
32    for (int i = 0; i <= size; i++) {
33        printf("%d ", array[i]);
34    }
```

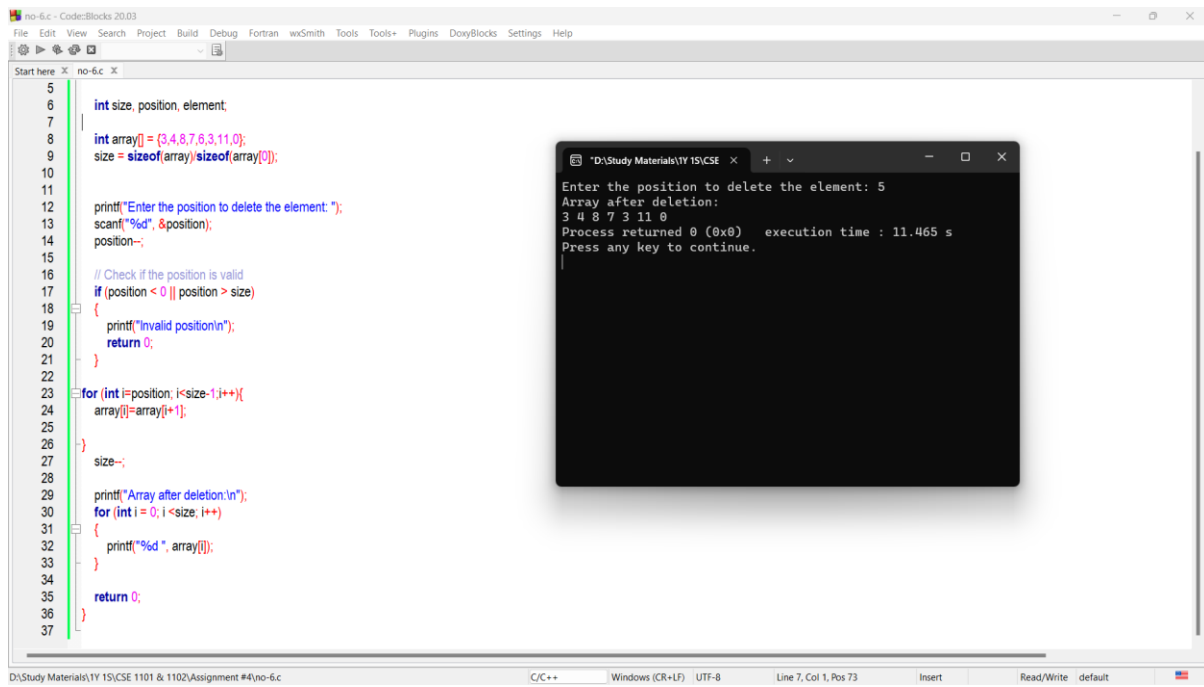
Enter the position to insert the element: 3
Enter the element to insert: 10
Array after insertion:
3 4 10 8 7 6 3 11 0
Process returned 0 (0x0) execution time : 6.295 s
Press any key to continue.

Problem No – 7

Title:

Write a program delete Element from an Array at a Desired or Specific Position.

Code and Output:



The screenshot displays a C++ IDE with a source code editor on the left and a terminal window on the right. The code in the editor defines an array of 8 integers: {3, 4, 8, 7, 6, 3, 11, 0}. It prompts the user to enter a position to delete an element. If the position is invalid (less than 0 or greater than or equal to the array size), it prints an error message. Otherwise, it shifts all elements from the specified position one index to the right and then prints the modified array. The terminal window shows the user entering '5', the resulting array '3 4 8 7 3 11 0', and the execution time of 11.465 seconds.

```
5
6 int size, position, element;
7
8 int array[] = {3,4,8,7,6,3,11,0};
9 size = sizeof(array)/sizeof(array[0]);
10
11
12 printf("Enter the position to delete the element: ");
13 scanf("%d", &position);
14 position--;
15
16 // Check if the position is valid
17 if (position < 0 || position >= size)
18 {
19     printf("Invalid position\n");
20     return 0;
21 }
22
23 for (int i=position; i<size-1; i++){
24     array[i]=array[i+1];
25 }
26 size--;
27
28 printf("Array after deletion:\n");
29 for (int i = 0; i < size; i++)
30 {
31     printf("%d ", array[i]);
32 }
33
34 return 0;
35
36
37
```

```
*D:\Study Materials\TY 15\CSE
Enter the position to delete the element: 5
Array after deletion:
3 4 8 7 3 11 0
Process returned 0 (0x0)   execution time : 11.465 s
Press any key to continue.
```