Programming Fundamentals

Lab Report

Lab 10

Group Members Name & Reg #:	**Muhammad Haris Irfan** **(FA18-BCE-090)**
Class	Programming Fundamentals CSC103 (**BCE-2B**)
Instructor's Name	Dilshad Sabir

In Lab Tasks

Question no: 1

•Your task is to declare a 2D array whose dimensions should be entered by the user of the program.

•Then you should initialize the array with ones (using nested loops).

•Next you have to write a function array_multiply() which takes in the array or its pointer as argument and multiplies all its entries with a user input number. (Hint: you will also need to pass in the dimensions of the matrix to this function).

•Similarly write a function array_add() that adds a constant number to all the entries in a 2D array.

•Print the results of calling these functions.

Solution:

The Code for the following code is attached below,

```
3
4   int array_add(int * ptr,int m,int n)
5   {
6       int a;
7       printf("Enter the value to be add?");
8       scanf("%d",&a);
9
10      for(int i=0;i<m;i++)
11      {
12          for(int j=0;j<n;j++)
13          {
14              *(ptr+(i*n)+j)+=a;
15          }
16      }
17  }22
18
19  int array_mul(int * ptr,int m,int n)
20  {
21      int a;
22      printf("Enter the value to be multiplied?");
23      scanf("%d",&a);
24
25      for(int i=0;i<m;i++)
26      {
27          for(int j=0;j<n;j++)
28          {
29              *(ptr+(i*n)+j)*=a;
30          }
31      }
32  }
```

```c
int main()
{
     int m,n,i,j;



    printf("Enter the Number of Rows?");
    scanf("%d",&m);

    printf("Enter the Number of Columns?");
    scanf("%d",&n);

    int arr[m][n];


    for(i=0 ; i<m ;i++)
    {
        for(j=0 ; j<n ;j++)
        {
            arr[i][j]=1;
        }
    }
array_add(arr,m,n);

 for (i = 0; i < m; i++)
    {
        for (j = 0; j < n; j++)
        {
            printf("%d    ", arr[i][j]);
        }
        printf(" \n");
```

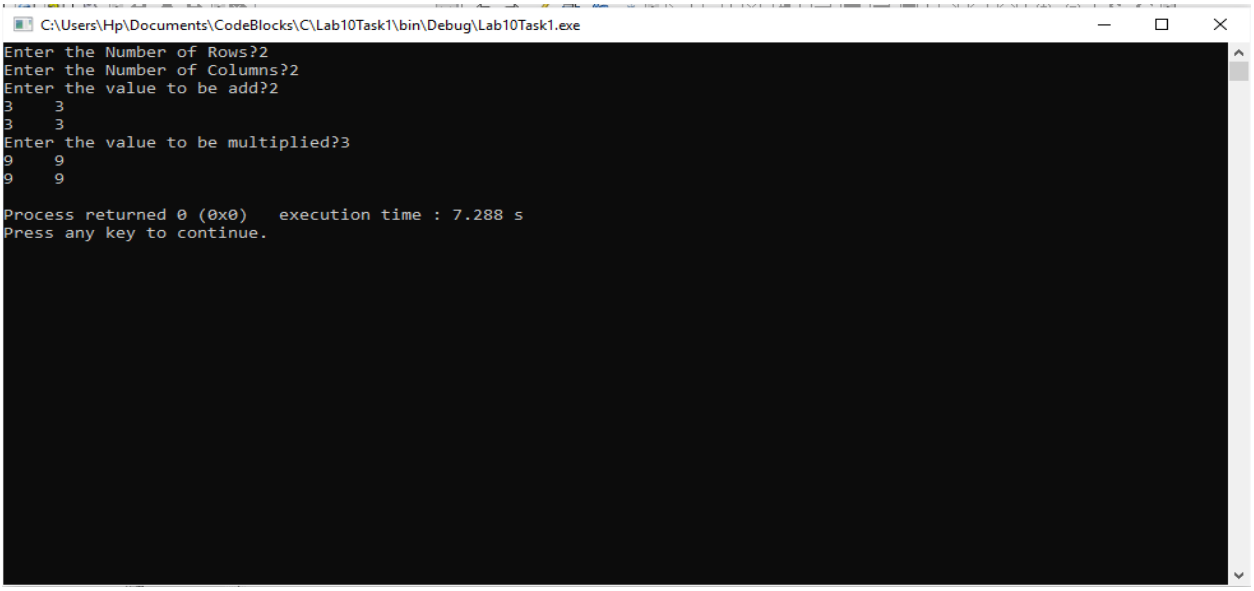
```c
        printf(" \n");
    }

    array_mul(arr,m,n);

for (i = 0; i < m; i++)
    {
        for (j = 0; j < n; j++)
        {
            printf("%d    ", arr[i][j]);
        }
        printf(" \n");
    }


    return 0;
}
```

The Result of the following code is attached below:



The output result verifies that our code is correct.

Question no: 2(A)

In this task you have to make a magic square and display it on the screen. You are given a Starter

Code (Annex I), that does the following:

•Asks the user to enter the order ‘n’ of the magic square (odd numbers only).

•Declares a 2D array of size n x n and initializes it with zeros.

•Prints the magic square on the screen.

Your job is to complete this code by implement the algorithm discussed in the Reading Task 2.

Solution

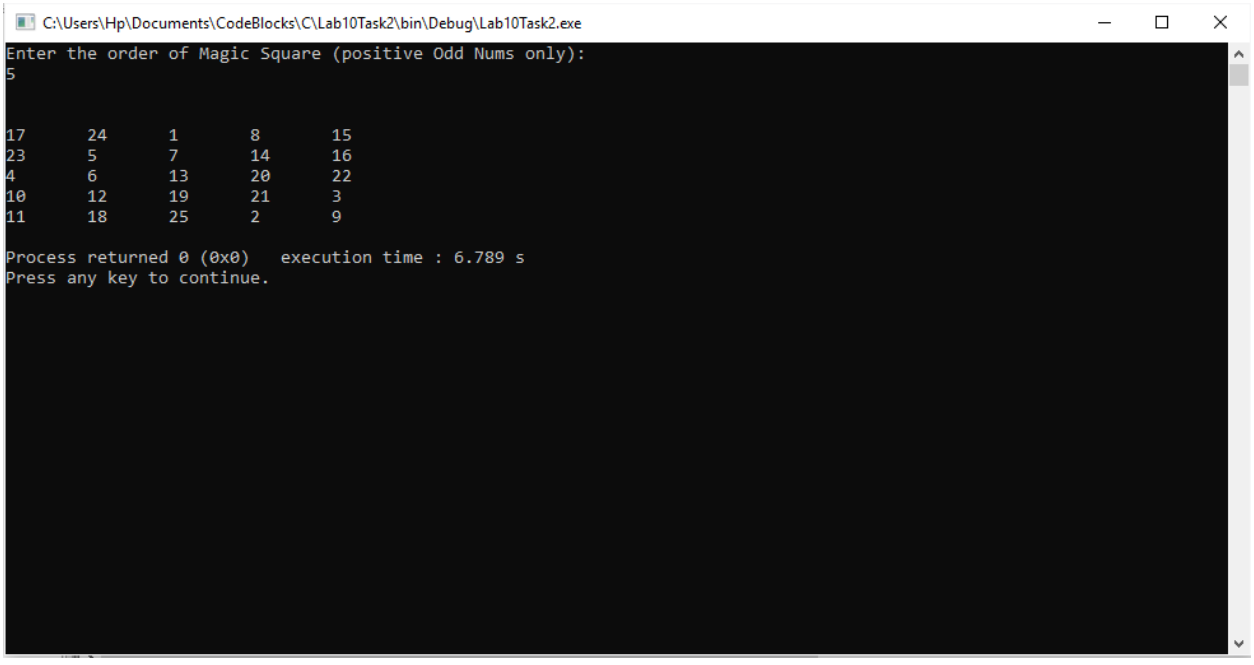
The code is shown below for the given program and its results are given below,

```
40
41      /************ INSERT YOUR CODE HERE ********/
42
43      col_t=col;
44      row_t=row;
45     int cornor =0;
46    for(int a=0; a<n ;a++)
47    {
48
49        for(int b=0; b<n;b++)
50        {
51             if(row_t==0 && col_t!=n-1)
52       {
53           row_t=row_t+(n-1);
54           col_t=col_t+1;
55
56
57       }
58       else if (row_t==0 && col_t==n-1)
59       {
60           row_t=n-1;
61           col_t=0;
62
63            if(mSquare[row_t][col_t]!=0)
64            {
65                row_t=row_t-(n-2);
66                col_t=col_t+(n-1);
67
68            }
69       }
70       else if(row_t!=0 && col_t==n-1)
71       {
72           row_t=row_t-1;
73           col_t=0;
74
75
76       }
77
78         else
79        {
80            row_t=row_t-1;
81
82       if(col_t==n-1)
83       {
84           col_t=0;
85       }
86       else{
87
88          col_t=col_t+1;
89       }
90        }
91
92       count=count+1;
93
94      if(mSquare[row_t][col_t]==0)
95      {
96    mSquare[row_t][col_t] = count ;
97      }
98
99     else
100    {
101        row_t=row_t+2;
102        col_t=col_t-1;
103        mSquare[row_t][col_t] = count ;
104    }
105
106
```

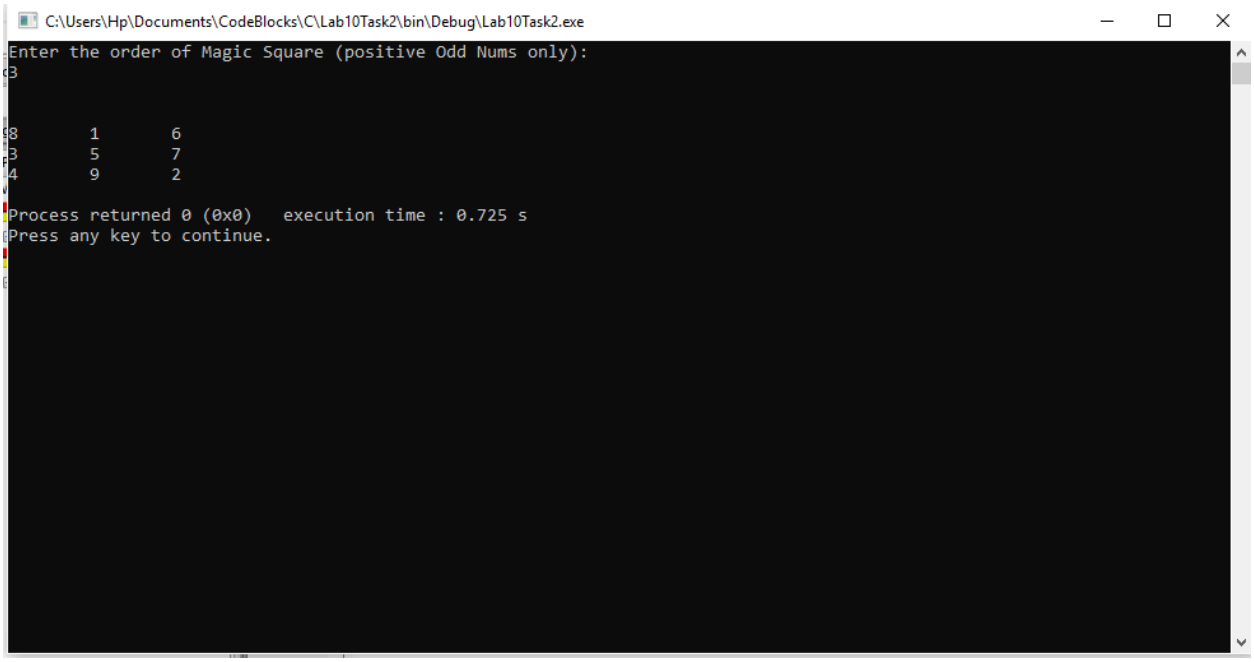
```
105
106
107     }
108     }
109      for(int i=0; i<n; i++)
110         {
111              for(int j=0; j<n; j++)
112              {
113                   printf("%d\t", mSquare[i][j]);
114              }
115              printf("\n");
116         }
117
118         if(count==(n*n)-1)
119        {
120             return 0;
121         }
122
123
124         return(0);
125         }
126
127
128
```

This code fills the magic square for odd numbers.

For n=5



For n=3



Our program works correctly for all entered odd values, hence our program is correct..

THE END

*there was no post lab for this lab.