**Programming Fundamentals**

**Lab 10**

**Submitted To:**

Mr. Dilshad Sabir

**Submitted By:**

Manaal Waseem

FA18-BCE-074

**In Lab:**

**Task 1:**

**2D Arrays and user defined functions**

1. **Your task is to declare a 2D array whose dimensions should be entered by the user of the**
2. **program.**
3. **Then you should initialize the array with ones (using nested loops).**
4. **Next you have to write a function array\_multiply() which takes in the array or its pointer as**
5. **argument and multiplies all its entries with a user input number. (Hint: you will also need to**
6. **pass in the dimensions of the matrix to this function).**
7. **Similarly write a function array\_add() that adds a constant number to all the entries in a 2D**
8. **array.**
9. **Print the results of calling these functions.**

**Program:** In this program, a variable length array is declared. The array is initialized with ones. In functions, **“array\_multiply()”** and **“array\_add()”**, a constant is multiplied and added to all entries of the declared array, respectively. The array is passed from main function to these functions using pointers.

1 #include <stdio.h>

2 #include <stdlib.h>

3

4 **void** array\_multiply(**int** \*ptr\_arr,**int** n, **int** r, **int** c);

5 **void** array\_add(**int** \*ptr\_arr, **int** n, **int** r, **int** c);

6

7 **int** main()

8 {

9 **int** num;

10 **int** row;

11 **int** cols;

12

13

14 printf("Enter the dimensions for a matrix(rows\*columns)");

15 scanf("%d %d",&row,&cols);

16

17 **int** array[row][cols];

18

19 **int** \*ptr\_array=&array[0][0];

20

21 **for**(**int** r=0; r<row; r++)

22 {

23 **for**(**int** c=0; c<cols; c++)

24 {

25 array[r][c]=1;

26 }

27 }

28

29 **for**(**int** r=0; r<row; r++)

30 {

31 **for**(**int** c=0; c<cols; c++)

32 {

33 printf("%d\t",array[r][c]);

34 }

35 printf("\n");

36 }

37

38 printf("\n");

39

40 printf("Enter a number: ");

41 scanf("%d",&num);

42

43 printf("\n");

44

45 array\_multiply(ptr\_array,num,row,cols);

46

47 printf("\n");

48

49 array\_add(ptr\_array,num,row,cols);

50

51 **return** 0;

52 }

53

54 **void** array\_multiply(**int** \*ptr\_arr, **int** n, **int** r, **int** c)

55 {

56 **for**(**int** i=0; i<r; i++)

57 {

58 **for**(**int** j=0; j<c; j++)

59 {

60 printf("%d\t",n\*(\*((ptr\_arr+i\*c)+j)));

61 }

62 printf("\n");

63 }

64

65 }

66

67 **void** array\_add(**int** \*ptr\_arr,**int** n, **int** r, **int** c)

68 {

69 **for**(**int** k=0; k<r; k++)

70 {

71 **for**(**int** l=0; l<c; l++)

72 {

73 printf("%d\t",n+(\*((ptr\_arr+k\*c)+l)));

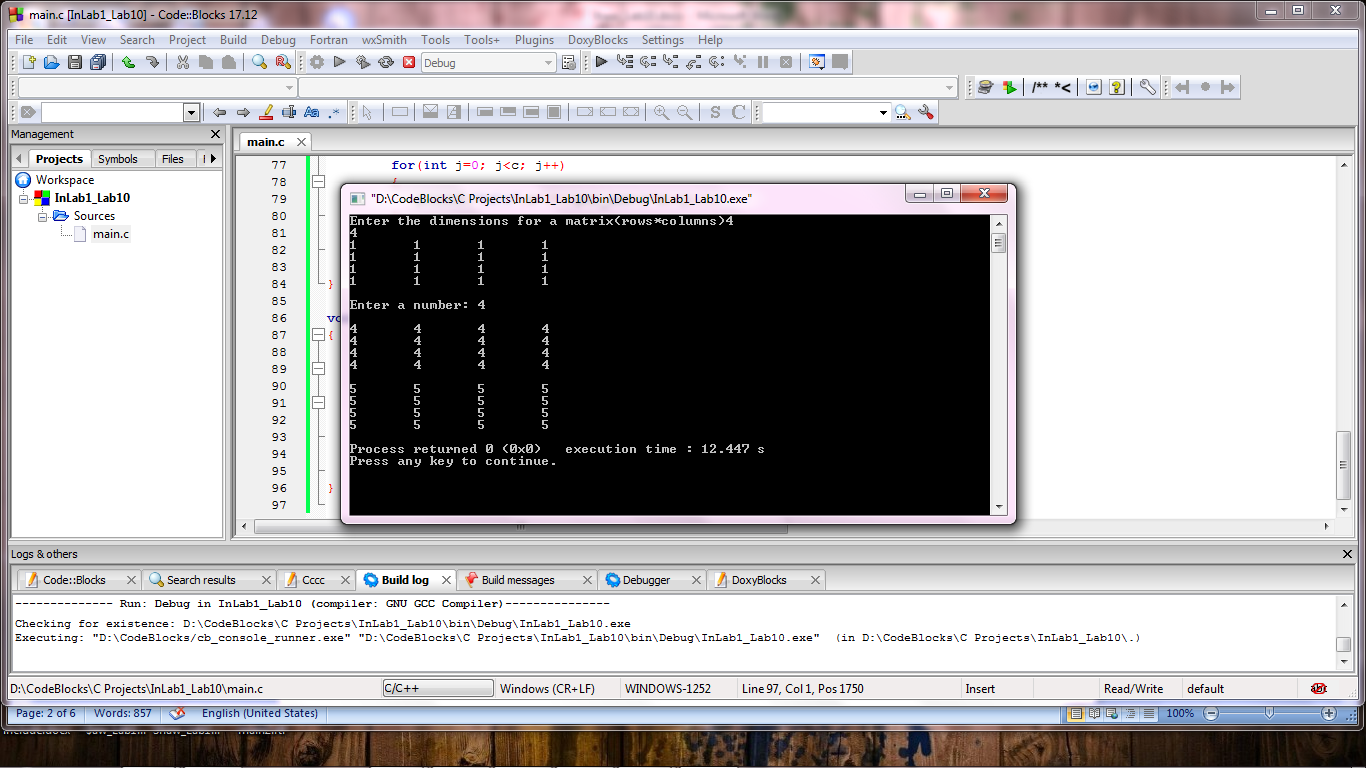
74 }

75 printf("\n");

76 }

77 }

**Output:**



**Task 2:**

**Implementing the algorithm for the Magic Square**

**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.**

Code Listing 2

#include <stdio.h>

#include <stdlib.h>

int magic\_square(int n);

int main()

{

int n;

printf("Enter the order of Magic Square (positive Odd Nums only): \n");

scanf("%d", &n);

printf("\n\n");

if(magic\_square(n) != 0)

printf("\nPlease enter correct value for n!!");

return 0;

}

int magic\_square(int n)

{

/// to rule out zero, negative and even inputs

if((n<1) || (n%2 == 0))

return(-1);

/// Declare an n x n array using C VLAs (Variable Length Arrays)

int mSquare[n][n];

/// Initialize the matrix with zeros

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

{

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

{

mSquare[i][j] = 0;

}

}

int count = 1; /// The first entry in the Magic Square

///start from middle col in the first row.

int col = (n-1)/2;

int row = 0;

/// Step1: Fill 1 in the middle column first row

mSquare[row][col] = count;

int row\_t; /// for temporary storage

int col\_t;

/\*\*\*\*\*\*\*\*\*\*\*\* INSERT YOUR CODE HERE \*\*\*\*\*\*\*\*/

/// Print the Magic Square

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

{

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

{

printf("%d\t", mSquare[i][j]);

}

printf("\n");

}

return(0);

}

**Program:** In this program, the logic for magic square has designed according to the algorithm given in the reading material. Hence, the magic square is printed as output on the console.

1 #include <stdio.h>

2 #include <stdlib.h>

3 **int** magic\_square(**int** n);

4

5 **int** main()

6 {

7 **int** n;

8 printf("Enter the order of Magic Square (positive Odd Nums only): \n");

9 scanf("%d", &n);

10 printf("\n\n");

11 **if**(magic\_square(n) != 0)

12 printf("\nPlease enter correct value for n!!");

13 **return** 0;

14 }

15

16 **int** magic\_square(**int** n)

17 {

18 **/// to rule out zero, negative and even inputs**

19 **if**((n<1) || (n%2 == 0))

20 **return**(-1);

21 **/// Declare an n x n array using C VLAs (Variable Length Arrays)**

22 **int** mSquare[n][n];

23 **/// Initialize the matrix with zeros**

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

25 {

26 **for**(**int** j=0; j<n; j++)

27 {

28 mSquare[i][j] = 0;

29 }

30 }

31 **int** count = 1; **/// The first entry in the Magic Square**

32 **///start from middle col in the first row.**

33 **int** col= (n-1)/2;

34 **int** row = 0;

35 **/// Step1: Fill 1 in the middle column first row**

36 mSquare[row][col] = count;

37

38 **int** row\_t; **/// for temporary storage**

39 **int** col\_t;

40

41 **/\*\*\*\*\*\*\*\*\*\*\*\* INSERT YOUR CODE HERE \*\*\*\*\*\*\*\*/**

42

43 **/// fundamental move**

44 --row;

45 ++col;

46

47 **for** (count=2; count <= n\*n;)

48 {

49 **/// if fundamental move leaves the matrix it is**

50 **/// wrapped to lasr row or first column**

51 **if** (row < 0)

52 row = n-1;

53

54 **if** (col >= n)

55 col = 0;

56

57 **if** (mSquare[row][col]) **/// if a square is filled**

58 {

59 col = col\_t;

60 row = ++row\_t;

61 }

62 **else**

63 {

64 mSquare[row][col] = count++; **///set number**

65 row\_t = row;

66 col\_t = col;

67 **/// fundamental move**

68 col++;

69 row--;

70 }

71

72 }

73

74 **/// Print the Magic Square**

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

76 {

77 **for**(**int** j=0; j<n; j++)

78 {

79 printf("%d\t", mSquare[i][j]);

80 }

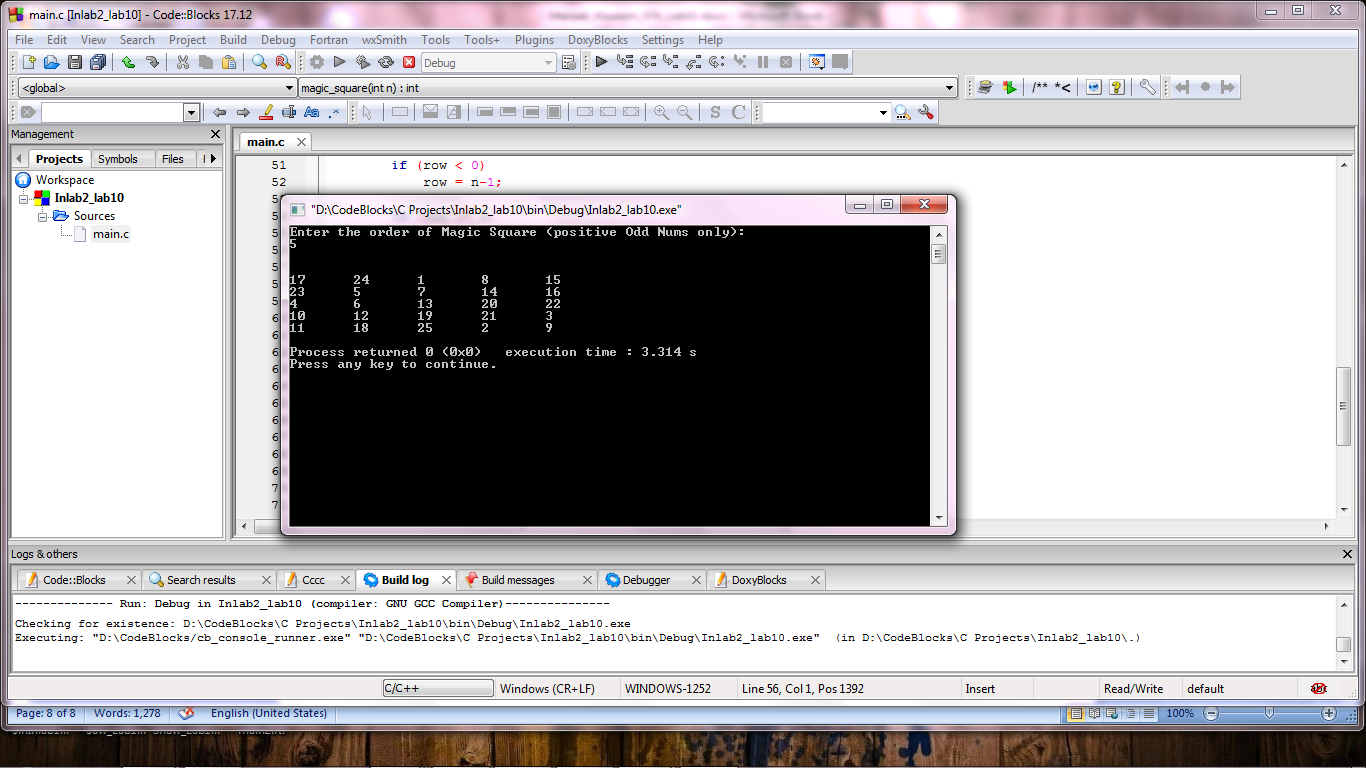
81

82 printf("\n");

83 }

84 **return**(0); }

**Output:**



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**THE END**