JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY, GUNA
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Course: Computer Programming Lab
Course Code: CS201
B. Tech. I Sem. (CSE, ECE, MECH, CE, CHE)

Lab-10

Aim: Printing of Patterns

Patterns: These are defined as a design that repeats a specific number of times, especially a design made from repeated lines, shapes, or colors on a surface. Patterns exist all around us. We can find a pattern in the trees, in the window frames, on the floor, in our clothes, etc. One such real-life example is the zebra pattern. Patterns can be finite and infinite. A pattern program helps to improve the concept of looping and algorithms.

In this lab, various patterns consisting of numbers, alphabets or symbols in a particular form will be displayed in C using the loop statements. They can be easily displayed with the help of **nested loops** where the **outer loop** controls the number of **rows**, and the **inner loop** controls the **(column)** data in each row containing the contents to be printed. Looping can be done using any of the loop statement, but writing programs using **for** loop is more straightforward than the **while/do-while** loop.

Note: No standard formula/method is available to print all types of patterns. All required input/output parameters, test conditions, print values and initializations are taken according to particular pattern problem. However, following procedure can be applied to print most of the patterns:

- Pattern programs can have either two user inputs (number of rows and columns) or only one user input (either number row or column. Preferably, number of rows is taken as input and number of columns is defined in terms of rows).
- Every pattern program needs **at least two nested loops**.
- Following template can be used as default template for printing most of the patterns :

```
#include<stdio.h>
int main ( )
{
int row, col, i, j;
printf (“Enter number of rows and columns\n”);
scanf (“%d %d”, &row, &col );
for ( i = 1; i <= rows;  i++)        //Controls rows and executes lesser number of times than inner loop
    {
    for ( j = 1; j <= col;  j++)     //Controls columns and executes more number of times than outer loop
            {
            pattern statements;
            }
    printf (“\n”);                   //Moves curser in the next row
    }
return (0);
}
```

- If number of columns (j) of a pattern are varying with its number of rows (i), then the relationship between columns and rows can be written as:

$$j = j_0 + (i - i_0)d$$

$where\ i_0 = initial\ row\ number;\ j_0 = initial\ column\ number;$
$d = difference\ between\ conscutive\ column\ numbers\ in\ the\ slope$

In the same way, if the values (v) of a pattern are varying with columns (j), then the relationship between values and column can be written as:

$$v = v_0 + (j - j_0)d$$

$where\ v_0 = initial\ pattern\ value\ in\ the\ initial\ column$
$d = difference\ between\ conscutive\ values\ in\ the\ column$

Replace j with i in the above expression if pattern values are varying with pattern rows. **This formula is not applicable if pattern values are varying with rows and columns both.**

Example#1: Write a c-language program to print a pattern shown in figure which consists of two parts, ten rows, five columns and two slopes.

First part (from row number 1 to 5) of pattern has constant pattern values (*) and a slope in which number of columns (initial column $j_0 = 1$) are varying with rows (initial row $i_0 = 1$). Therefore, for upper part of the pattern relationship between columns and rows can be written as:

$$j = 1 + (i - 1) \times (1) = i$$

It means, pattern value (*) will be printed only for $\boldsymbol{j \geq i}$ elsewhere space will be printed.

In second part (from row number 6 to 10) of the pattern, number of columns (initial column $j_0 = 5$) are varying with rows (initial row $i_0 = 6$), due to slope and display values (initial value $v_0 = 1$), are also varying with columns (initial column $j_0 = 5$) Therefore, relationship between columns and rows can be derived as:

$$j = 5 + (i - 6) \times (-1) = 11 - i = row + 1 - i$$

It means, pattern values will be printed only for $\boldsymbol{j \geq row + 1 - i}$ elsewhere space will be printed.

Similarly, relationship between pattern values and column can be derived as:

$$v = 1 + (j - 5) \times (-1) = 6 - j = col + 1 - j$$

```
*****
****
***
**
*
1
21
321
4321
54321
```

Program using **two for loops** and **if-else**	Program using **for loops** only

#include<stdio.h>
int main ( )
{
int row, col, i, j;
printf ("Enter number of rows and columns\n");
scanf ("%d %d", &row, &col);
for (i = 1; i <= row; i++)
{
for (j = 1; j <= col; j++)
{
if (i <= row/2) //to print upper part of pattern
{
if (j >= i)
printf ("*");
else
printf (" ");
}
else //to print lower part of pattern
{
if (j >= row+1-i)
printf ("%d", col+1-j);
else
printf (" ");
}
}
printf ("\n");
}
return (0);
}
	#include<stdio.h>
	int main ( )
	{
	int row, col, i, j;
	printf ("Enter number of rows and columns\n");
	scanf ("%d %d", &row, &col);
	
	//To print upper half of pattern
	
	for (i = 1; i <= row/2; i++)
	{
	for (j=1; j < i; j++)
	printf (" ");
	for (j=col; j >= i; j--)
	printf ("*");
	printf ("\n");
	}
	//To print lower half of pattern
	
	for (i = (row/2)+1; i <= row; i++)
	{
	for (j = row+1-i; j > 1; j--)
	printf(" ");
	for (j = row+1-i; j <=col; j++)
	printf ("%d",col+1-j);
	printf ("\n");
	}
	return (0);
	}

Example#2: Write a c-language program to print a character pattern shown in the figure.

```
#include<stdio.h>
int main ( )
{  int row, col, i, j;
printf ("Enter number of rows and columns\n");
scanf ("%d %d", &row, &col );
for ( i = 1; i <= row; i++)    {
   for ( j = 1; j <= col; j++)
      if(j >= row+1-i  &&  j <= row-1+i)
        printf ("%c", 64+i);
      else
        printf(" ");
  printf("\n");                    }
return (0);
}
```

```
   A
  BBB
 CCCCC
DDDDDDD
EEEEEEEEE
```

For first turn of the week:

Exercise 1-4: Write C-language user input program to print following patterns:

+ ***+*** ***+*** +++++++ ***+*** ***+*** ***+***	***+111 ***+111 ***+111 +++++++ aaa+ZZZ aaa+ZZZ aaa+ZZZ
***+111 ***+222 ***+333 +++++++ aaa+ZZZ bbb+YYY ccc+XXX	***+*** * + * * + * +++++++ * + * * + * ***+***

For second turn of the week:

Exercise 5-8: Write C-language user input program to print following patterns:

a4321 bb321 ccc21 dddd1 eeeee	1 232 34543 4567654 34543 232 1
********** *** *** * * 5 1 444 222 3333333333	0123456789 9012345678 9801234567 9870123456 9876012345 9876501234 9876540123 9876543012 9876543201

Practice Questions

(No need to include in your Practical Book)

Write C-language user input program to print following patterns: