**Batch: B2 Roll No.: 121**

**Experiment / assignment / tutorial No. 4**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

|  |
| --- |
| **TITLE:**  Program to print patterns |

**AIM:** Program to print patterns for ‘n’ rows using nested loop **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected OUTCOME of Experiment:**

CO2: Apply basic concepts of C programming for problem solving.

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

**Books/ Journals/ Websites referred:**

1. Programming in C, second edition, Pradeep Dey and Manas Ghosh, Oxford University Press.
2. Programming in ANSI C, fifth edition, E Balagurusamy, Tata McGraw Hill.
3. Introduction to programming and problem solving , G. Michael Schneider ,Wiley India edition.
4. [**http://cse.iitkgp.ac.in/~rkumar/pds-vlab/**](http://cse.iitkgp.ac.in/~rkumar/pds-vlab/)

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

**Problem Definition:**

The program is to print a pattern as given by the user. The program makes use of a nested loop to print a pattern of characters, numbers or alphabets.

**Example:**

Input: number of rows = 4, number of columns = 4

Output:

5

             4    5

         3   4   5

    2   3   4   5

1  2   3   4   5

**Algorithm:**

Pattern 1: To display an inverse star full triangle pattern

Step 1: Start

Step 2: Declare variables totalrow, row, space, symbol

Step 3: Read totalrow

Step 4: For loop. Initialise row ß totalrow. Repeat the steps if row >= 1. Decrement row by one after each iteration.

4.1: For loop. Initialize space ß 1. Repeat the steps if space <= totalrow – row. Increment space by 1 after each iteration.

4.1.1: Display a blank space

4.2: For loop. Initialize symbol ß 1. Repeat the steps if symbol <= 2\*row – 1. Increment symbol by 1 after each iteration.

4.2.1: Display a star symbol

4.2.2: Display a line break

Step 5: Stop

Pattern 2: To display a diamond pattern

Step 1: Start

Step 2: Declare variables totalrow, row, space, symbol, m, r\_n

Step 3: Read totalrow

Step 4: m ß totalrow/2 + 1

Step 5: r\_n ß totalrow – m

Step 6: For loop. Initialize row ß 1. Repeat the steps if row <= m. Increment row by 1 after each iteration.

6.1: For loop. Initialize space ß 1. Repeat the steps if space <= m – row. Increment space by 1 after each iteration.

6.1.1: Diaplay a blank space

6.2: For loop. Initialize symbol ß 1. Repeat the steps if symbol <= 2\*row – 1. Increment symbol by 1 after each iteration.

6.2.1: Display a star symbol

6.2.2: Display a line break

Step 7: For loop. Initialize row ß r\_n. Repeat the steps if row >= 1. Decrement row by 1 after each iteration.

7.1: For loop. Initialize space ß 0. Repeat the steps if space <= r\_n – row. Increment space by 1 after each iteration.

7.1.1: Display a blank space

7.2: For loop. Initialize symbol ß 1. Repeat the steps if symbol <= 2\*row – 1. Increment symbol by 1 after each iteration.

7.2.1: Display a star symbol

7.2.2: Display a line break

Step 8: Stop

Pattern 3: To display Pascal’s Triangle

Step 1: Start

Step 2: Declare variables totalrow, row, spaces, symbols, num.

Step 3: Initialize num ß 1

Step 4: Read totalrow

Step 5: For loop. Initialize row ß 0. Repeat the steps if row < totalrow. Increment row by 1 after each iteration

5.1: For loop. Initialize spaces ß 1. Repeat the steps if spaces <= totalrow – row. Increment spaces by 1 after each iteration.

5.1.1: Display two blank spaces

5.2: For loop: Initialize symbols ß 0. Repeat the steps if symbols <= row. Increment symbols by 1 after each iteration.

5.2.1: If symbols is 0 or row is 0

num ß 1

Else

num ß num\*(row – symbols + 1)/symbols

5.2.2: Display num

Step 6: Stop

Pattern 4: To display Floyd’s Triangle

Step 1: Start

Step 2: Declare variables i, j, k, n;

Step 3: Initialize k ß 1

Step 4: Read n

Step 5: For loop. Initialize i ß 0. Repeat the steps if i < n. Increment i by 1 after each iteration.

5.1: For loop. Initialize j = 0. Repeat the steps if j <= i. Increment j by 1 after each iteration

5.1.1: Display k

5.1.2: k ß k + 1

Step 6: Stop

Pattern 5: To display an Inverted Half-Triangle of alphabets in decreasing order

Step 1: Start

Step 2: Declare variables i, j, ch, n;

Step 3: Initialize ch ß 69

Step 4: Read n

Step 5: For loop. Initialize i ß n. Repeat the steps if i >= 1. Decrement i by 1 after each iteration.

5.1: For loop. Initialize j ß 1. Repeat the steps if j <= i. Increment j by 1 after each iteration.

5.1.1: Display ch using %c specifier

5.1.2: ch ß ch – 1

Step 6: Stop

**Implementation details:**

Pattern 1:

#include<stdio.h>

int main()

{

int totalrow, row, space, symbol;

printf("Enter the total number of rows: ");

scanf("%d", &totalrow);

for(row = totalrow; row >= 1; row--)

{

//print the spaces

for(space = 1; space <= totalrow - row; space++)

printf(" ");

//print the symbols

for(symbol = 1; symbol <= (2\*row) - 1; symbol++)

printf("\*");

printf("\n");

}

return 0;

}

Pattern 2:

#include<stdio.h>

int main()

{

int totalrow, row, space, symbol;

int m, r\_n;

printf("Enter the total number of rows: ");

scanf("%d", &totalrow); // totalrows = 9

m = totalrow/2 + 1; //mid m = 9/2 + 1 = 5

r\_n = totalrow - m; //rows after mid, r\_n = 9 - 5 = 4

for(row = 1; row <= m; row++) //ascending upto mid

{

for(space = 1; space <= (m - row); space++)

printf(" ");

for(symbol = 1; symbol <= (2\*row - 1); symbol++)

printf("\*");

printf("\n");

}

for(row = r\_n; row >= 1; row--)

{

for(space = 0; space <= (r\_n - row); space++)

printf(" ");

for(symbol = 1; symbol <= (2\*row - 1); symbol++)

printf("\*");

printf("\n");

}

return 0;

}

Pattern 3:

#include<stdio.h>

int main()

{

int totalrow, spaces, symbols, row, num = 1;

printf("No of rows: ");

scanf("%d", &totalrow);

for(row = 0; row < totalrow; row++)

{

//print spaces

for(spaces = 1; spaces <= (totalrow-row); spaces++)

printf(" ");

//print numbers

for(symbols = 0; symbols <= row; symbols++)

{

if(symbols == 0 || row == 0)

num = 1;

else

num = num\*(row - symbols + 1)/symbols;

printf("%4d", num);

}

printf("\n");

}

return 0;

}

Pattern 4:

#include<stdio.h>

int main()

{

int i, j, n, k;

k = 1;

printf("Enter the number of rows of the\nFloyd's triangle to be printed:");

scanf("%d", &n);

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

{

for(j = 1; j <= i; j++)

{

printf(" %d", k);

k++;

}

printf("\n");

}

return 0;

}

Pattern 5:

#include<stdio.h>

int main()

{

int i, j, n, ch;

printf("\nEnter the number of rows of the\nInverted Alphabet Triangle to be printed: ");

scanf("%d", &n);

for(i = n; i >= 1; i--)

{

ch = 69;

for(j = 1; j <= i; j++)

{

printf("%c", ch);

ch--;

}

printf("\n");

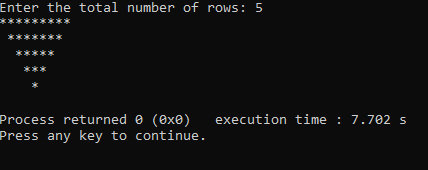
}

return 0;

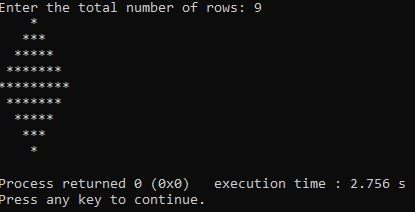
}

**Output(s):**

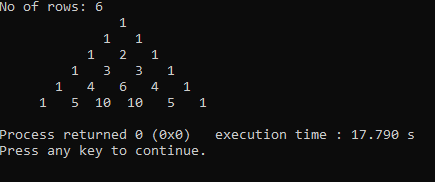
Pattern 1: Inverted star pattern triangle



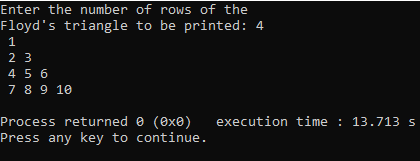
Pattern 2: Diamond star pattern



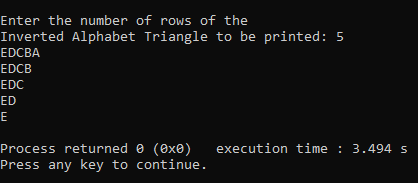
Pattern 3: Pascal’s Triangle



Pattern 4: Floyd’s Triangle



Pattern 5: Inverted Half-Triangle of alphabets in decreasing order



**Conclusion:**

Thus, in this experiment, the basic conscepts of C programming have been applied for problem solving. The problem solved here was related to patterns, which may have mathematical significance. The patterns have been generated by using nested for loops. Thus, it can be concluded that using nested for loops, patterns and other complex figures can be displayed on the output screen with minimal effort and more efficiency in coding as well.

**Post Lab Descriptive Questions**

Write a program to print the following:

1

2 4

3 6 9

4 8 12 16

5 10 15 20 25

6 12 18 24 30 36

7 14 21 28 35 42 49

8 16 24 32 40 48 56 64

9 18 27 36 45 54 63 72 81

10 20 30 40 50 60 70 80 90 100

#include<stdio.h>

int main()

{

int i, j, n, k;

printf("Enter the number of rows of the Multipli-\ncation Table Half-triangle to be printed: ");

scanf("%d", &n);

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

{

k = i;

for(j = 1; j <= i; j++)

{

k = k\*j;

printf(" %d", k);

k = i;

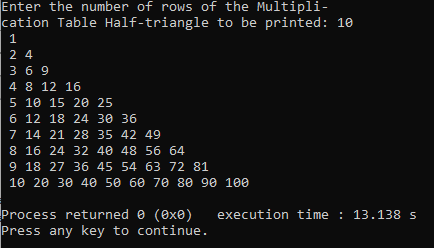
}

printf("\n");

}

return 0;

}



**Date: \_\_12-12-2021\_\_\_ Signature of faculty in-charge**