Coding Ability

1. **C Program to check if two given matrices are identical**

[code language=”cpp”]
#include
#define N 4

// This function returns 1 if A[][] and B[][] are identical
// otherwise returns 0
int areSame(int A[][N], int B[][N])
{
int i, j;
for (i = 0; i < N; i++)
for (j = 0; j < N; j++)
if (A[i][j] != B[i][j])
return 0;
return 1;
}

int main()
{
int A[N][N] = { {1, 1, 1, 1},
{2, 2, 2, 2},
{3, 3, 3, 3},
{4, 4, 4, 4}};

int B[N][N] = { {1, 1, 1, 1},
{2, 2, 2, 2},
{3, 3, 3, 3},
{4, 4, 4, 4}};

if (areSame(A, B))
printf(“Matrices are identical”);
else
printf(“Matrices are not identical”);
return 0;
}
[/code]

2. **Print a given matrix in spiral form**

   Ques: Given a 2D array, print it in spiral form. See the following examples.

   NOTE:- Please comment down the code in other languages as well below .

   Input:
   1 2 3 4
   5 6 7 8
   9 10 11 12
   13 14 15 16
   Output:
   1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10
   Input:
   1 2 3 4 5 6
   7 8 9 10 11 12
   13 14 15 16 17 18
   Output:
   1 2 3 4 5 6 12 18 17 16 15 14 13 7 8 9 10 11

Solution: Using C++

```
#include <bits/stdc++.h>
using namespace std;
#define R 3
#define C 6

void spiralPrint(int m, int n, int a[R][C])
{
    int i, k = 0, l = 0;

    /* k - starting row index
        m - ending row index
        l - starting column index
        n - ending column index
        i - iterator
```

```
*/

while (k < m && l < n) {
        /* Print the first row from
                the remaining rows */
        for (i = l; i < n; ++i) {
                cout << a[k][i] << " ";
        }
        k++;

        /* Print the last column
        from the remaining columns */
        for (i = k; i < m; ++i) {
                cout << a[i][n - 1] << " ";
        }
        n--;

        /* Print the last row from
                        the remaining rows */
        if (k < m) {
                for (i = n - 1; i >= l; --i) {
                        cout << a[m - 1][i] << " ";
                }
                m--;
        }

        /* Print the first column from
                        the remaining columns */
        if (l < n) {
                for (i = m - 1; i >= k; --i) {
```

```
                cout << a[i][l] << " ";
            }
            l++;
        }
    }
}

/* Driver program to test above functions */
int main()
{
    int a[R][C] = { { 1, 2, 3, 4, 5, 6 },
                    { 7, 8, 9, 10, 11, 12 },
                    { 13, 14, 15, 16, 17, 18 } };

    spiralPrint(R, C, a);
    return 0;
}
```

3. **Ques: Given an n-by-n matrix of 0's and 1's where all 1's in each row come before all 0's, find the most efficient way to return the row with the maximum number of 0's.**

   {1,1,1,1},
   {1,1,0,0},
   {1,0,0,0},
   {1,1,0,0},

Solution:

```
#include
#include
#include
#define COL 4
#define ROW 4
using namespace std;
int main()
{
int arr[ROW][COL]= {
{1,1,1,1},
{1,1,0,0},
{1,0,0,0},
{1,1,0,0},
};

int rownum;
int i = 0, j = COL-1;
while(i;0)
{
if(arr[i][j]==0)
{
j–;
rownum=i;}
else
i++;
}
printf(“%d”,rownum);
getch();
return 0;
}
```

4. **Ques: A Pythagorean triplet is a set of three integers a, b and c such that $a^2 + b^2 = c^2$. Given a limit, generate all Pythagorean Triples with values smaller than given limit.**

Input : limit = 20
Output : 3 4 5
8 6 10
5 12 13
15 8 17
12 16 20

A **Simple Solution** is to generate these triplets smaller than given limit using three nested loop. For every triplet, check if Pythagorean condition is true, if true, then print the triplet. Time complexity of this solution is $O(limit^3)$ where 'limit' is given limit.

An **Efficient Solution** can print all triplets in O(k) time where k is number of triplets printed. The idea is to use square sum relation of Pythagorean triplet, i.e., addition of squares of a and b is equal to square of c, we can write these number in terms of m and n such that,

$a = m^2 - n^2$
$b = 2 * m * n$
$c = m^2 + n^2$

because,

$a^2 = m^4 + n^4 – 2 * m^2 * n^2$
$b^2 = 4 * m^2 * n^2$
$c^2 = m^4 + n^4 + 2* m^2 * n^2$

We can see that $a^2 + b^2 = c^2$, so instead of iterating for a, b and c we can iterate for m and n and can generate these triplets.

Below is C implementation of above idea.

[code language="cpp"]

```
//  A C program to generate pythagorean triplets
// smaller than a given limit
#include <stdio.h>
#include <math.h>

//  Function to generate pythagorean triplets
//  smaller than limit
void pythagoreanTriplets(int limit)
{
// triplet:  a^2 + b^2 = c^2
int a, b, c=0;

//  loop from 2 to max_limitit
int m = 2;
```

```
// Limiting c would limit all a, b and c
while (c < limit)
{
// now loop on j from 1 to i-1
for (int n = 1; n < m; ++n)
{
// Evaluate and print triplets using
// the relation between a, b and c
a = m*m – n*n;
b = 2*m*n;
c = m*m + n*n;

if (c > limit)
break;

printf("%d %d %d\n", a, b, c);
}
m++;
}
}

// Driver program
int main()
{
int limit = 20;
pythagoreanTriplets(limit);
return 0;
}
[/code]
```

Output:

```
3 4 5
8 6 10
5 12 13
15 8 17
12 16 20
```

Time complexity of this approach is O(k) where k is number of triplets printed for a given limit (We iterate for m and n only and every iteration prints a triplet)

5. **C program to calculate the GCD(Greatest Common Divisor) of two numbers**
   In this program, we are required to find the greatest common divisor of any two given numbers that are input by the user.
   The logic of the following program is to subtract the smaller number from the larger number.
   The obtained result is then assigned to the variable holding the larger of the two numbers.
   The process is repeated until num1 and num2 are equal.

   Solution:

```
#include <stdio.h>

int main()
{
    int num1, num2;

    printf("Enter the first number: ");
    scanf("%d",&num1);
    printf("Enter the second number: ");
    scanf("%d",&num2);
    while(num1!=num2)
    {
        if(num1 > num2)
            num1 -= num2;
        else
            num2 -= num1;
    }
    printf("The GCD of the given numbers = %d",num1);
    return 0;
}
```

6. **C Program to find the GCD of Three Numbers**
   In this case we have to find the GCD of three numbers.
   To calculate the GCD, we have to first find the greatest of the three numbers.
   Thereafter, we find the common number that when divided by each of the three numbers gives a remainder 0.

Solution:

```
#include <stdio.h>
int greatest_num(int a, int b, int c)
{
if(a>=b && a>=c)
{
return a;
}

else if(b>=a && b>=c)
{
return b;
}

else if(c>=a && c>=b)
{
return c;
}
}

int main()
{

int m,n,o;
printf("Enter the first number: ");
scanf("%d",&m);
printf("Enter the second number: ");
scanf("%d",&n);
printf("Enter the third number: ");
scanf("%d",&o);
int result;

for(result=greatest_num(m,n,o); result>=1; result--){
if(m%result==0 && n%result==0 && o%result==0){
break;
}
}
printf("GCD of the given numbers is : %d\n",result);
return 0;

}
```

7. **C Program to check if the given number is prime or not**
   Prime numbers can only be divided evenly by 1 or itself, so these numbers are prime numbers.
   A few examples of such numbers are 2 , 3 , 5 , 7 , 11 , 13 , ……. and so on.

   Solution:

```
#include<stdio.h>

int main ()
{
  int n, i, a = 0, count = 0;
  printf ("Enter the number to check if it is prime or not: ");
  scanf ("%d", &n);
  a = n / 2;
  for (i = 2; i <= a; i++)
  {
    if (n % i == 0)
          {
            printf ("The given number is not prime");
            count = 1;
            break;
          }
  }
  if (count == 0)
    printf ("The given number is prime");
  return 0;
}
```

8. **C Program to Sort first half in ascending order and second half in descending order in an Array**
   In this problem, we are required to display the first half of the array in ascending order and the remaining half in descending order.

   Solution:

```
#include <stdio.h>

int main()
{
int a[100];
int n, i, j, temp;

printf("Enter the size of list : ");
scanf("%d",&n);
```

```
for(i=0; i<n; i++)
{
scanf("%d",&a[i]);
}

for(i=0; i<n/2; i++)
{
for(j=0; j<n/2; j++)
{
if(a[i]<a[j])
{
temp = a[i];
a[i] = a[j];
a[j] = temp;
}
}
}

for(i=n/2; i<n; i++)
{
for(j=n/2; j<n; j++)
{
if(a[i]>a[j])
{
temp = a[i];
a[i] = a[j];
a[j] = temp;
}
}
}


printf("First Half in Ascending order : ");
for(i=0; i<n/2; i++)
{
printf("%d ",a[i]);
}
printf("\nSecond Half in Descending order : ");
for(i=n/2; i<n; i++)
{
printf("%d ",a[i]);
}

return 0;
}
```

9. PRINTING PATTERN:

****
****
****
****

PREREQUISITE:
Basic knowledge of C language and loops.

ALGORITHM:

Take number of rows as input from the user (length of side of the square) and store it in any variable (‘l ‘ in this case).

Run a loop ‘l ‘ number of times to iterate through the rows . From i=0 to i<l. The loop should be structured as for(i=0;i<l;i++) .

Run a nested loop inside the previous loop to iterate through the columns. From j=0 to j<l. The loop should be structured as for(j=0;j<l;j++) .

Print ‘*’ inside the nested loop to print ‘*’s in all the columns of a row.

Move to the next line by printing a new line. printf(“\n”) .

Code in C:

```
#include<stdio.h>
int main()
{
int i,j,l;    //declaring integers i,j for loops and l for number of rows
printf("Enter the number of rows/columns\n");  //Asking user for input
scanf("%d",&l);    //Taking the input for number of rows
for(int i=0;i<l;i++)  //Outer loop for number of rows
  {
    for(int j=0;j<l;j++)   //Inner loop for number of columns in each row
      {
        printf("*");     //Printing '*' in each column of a row.
      }
    printf("\n");    //Printing a new line after each row has been printed.
  }
}
```

10. PRINTING PATTERN:

****
* *
* *
****

PREREQUISITE:
Basic knowledge of C language and use of loops.

ALGORITHM:

Take number of rows/columns as input from the user (length of side of square) and store it in any variable (‘l’ in this case).

Run a loop ‘l’ number of times to iterate through all the rows. From i=0 to i<l. The loop should be structured as for(i=0; i<l ; i++).

Run a nested loop inside the main loop for printing stars . From j=0 to j<l. The loop should be structured as for(j=0 ; j<l : j++).

Inside the above loop print stars only if i=0 or i=l-1 or j=0 or j=l-1 in all other cases print a blank space.

Move to the next line by printing a new line. printf(“\n”).

Code in C:

```
#include<stdio.h>
int main()
{
int i,j,l;    //declaring integers i,j for loops and l for the number of rows
printf(" Enter the number of rows\n");   //Asking user for input
scanf("%d",&l);   //taking input for number of rows and saving in variable l
for(i=0;i<l;i++) //Outer loop for number of rows
  {
    for(j=0;j<l;j++) //Inner loop for printing stars in each column of a row
      {
        if(i==0 || i==l-1 || j==0 || j==l-1) // condition for printing stars
          {
            printf("*");   // printing stars
          }
        else               // else condition to print spaces
          {
            printf(" ");   //printing spaces
          }
      }
    printf("\n");       //Printing a new line after a row has been printed
  }
}
```

11. PRINTING PATTERN:

```
****
 ****
  ****
   ****
```

PREREQUISITE:
Basic knowledge of C language and use of loops.

ALGORITHM

Take the number of rows as input from the user (length of side of rhombus) and store it in any variable.(‘l‘ in this case).

Run a loop ‘l’ number of times to iterate through each of the rows. From i=0 to i<l. The loop should be structured as for(i=0 ; i<l : i++).

Run a nested loop inside the main loop to print the spaces before the rhombus. From j=0 to j<i. The loop should be structured as for(j=0; j<i ; j++).

Run another nested loop inside the main loop after the previous loop to print the stars in each column of a row. From j=0 to j<l. The loop should be structured as for(j=0 ; j<l ; j++).

Move to the next line by printing a new line . printf(“/n”).

Code in C:

```
#include<stdio.h>
int main()
{
int i,j,l; //declaring integer variables i,j for loops and l for number of rows/columns
printf("enter number of rows/columns\n"); //asking the user for input
scanf("%d",&l); //taking input from the user
for(i=0;i<l;i++) //loop controlling number of rows
  {
    for(j=0;j<i;j++) //inner loop for spaces
      {
        printf(" ");
      }
    for(j=0;j<l;j++) //inner loop for printing the stars in each column of a row
      {
        printf("*");
      }
    printf("\n"); // printing a new line after each row
  }
}
```

12. C Program to check if two given matrices are identical ?

```
#include
#define N 4

// This function returns 1 if A[][] and B[][] are identical
// otherwise returns 0
    int areSame(int A[][N], int B[][N])
        {
            int i, j;
            for (i = 0; i < N; i++)
                {

                    for (j = 0; j < N; j++)
                            {

                                if (A[i][j] != B[i][j])
                                return 0;
                                return 1;
                            }

                 }

        int main()
           {
              int A[N][N] = { {1, 1, 1, 1},  {2, 2, 2, 2}, {3, 3, 3, 3}, {4, 4, 4, 4}};

             int B[N][N] = { {1, 1, 1, 1}, {2, 2, 2, 2}, {3, 3, 3, 3}, {4, 4, 4, 4}};

                if (areSame(A, B))
                     printf("Matrices are identical");
                else
                   printf("Matrices are not identical");
             return 0;
           }
```

13. PRINTING PATTERN:

******
******
******
******

PREREQUISITE:
Basic knowledge of C language and use of loops.

ALGORITHM:

Take the number of rows and columns as input from the user (length and breadth of the rectangle) and store it in two different variables. (‘r’ and ‘c’ in this case)

Run a loop ‘r’ number of times to iterate through all the rows. From i=0 to i<r. The loop should be structured as for(i=0 ; i<r ; i++)

Run a nested loop ‘c’ times to iterate though each column of a row. From j=0 to j<c. The loop should be structured as for(j=0 ; j<c ; j++).

Inside the nested loop print star to print a star in each column of a row.

In the main loop print a new line to move to the next line.

CODE IN C:

```
#include<stdio.h>
int main()
{
int i,j,r,c; //declaring integers i,j for loops and r,c for number of rows and columns
printf("Enter the number of rows\n"); //asking user for number of rows
scanf("%d",&r); //saving number of rows in a variable r
printf("Enter the number of columns\n"); //asking user for number of columns
scanf("%d",&c); //saving number of columns in a variable c
  for(int i=0;i<r;i++) //outer loop for number of rows
    {
      for(int j=0;j<c;j++) //inner loop for number of columns
        {
          printf("*"); //printing stars in each column of a row
        }
      printf("\n"); //printing new line after a row
    }
}
```

14. Print Right Triangle Star Pattern

PRINTING PATTERN:

```
*
**
***
****
```

PREREQUISITE:

Basic knowledge of C language and use of loops.

ALGORITHM:

Take the number of rows as input from the user and store it in any variable.(‘r‘ in this case).

Run a loop ‘r’ number of times to iterate through each of the rows. From i=0 to i<r. The loop should be structured as for(i=0 ; i<r : i++).

Run a nested loop inside the main loop to print the stars in each row of the triangle. From j=0 to j<i. The loop should be structured as for(j=0; j<=i ; j++).

In the nested loop print star.

In the main loop print a new line.

Code in C:

```
#include<stdio.h>
int main()
{
int i,j,r;      //declaring integer variables i,j for loops and r for number of rows
printf("Enter the number of rows :\n");     //Asking user for input
scanf("%d",&r);     //taking number of rows and saving it in variable r
for(i=0;i<r;i++)     // loop for number of rows
  {
    for(j=0;j<=i;j++)      // loop for stars per each row
    {
      printf("*");       //printing stars
    }
    printf("\n");         // printing newline after each row
  }
}
```

15. Print Pyramid Star Pattern

PRINTING PATTERN:

```
    *
   ***
  *****
*******
```

PREREQUISITE:

Basic knowledge of C language and use of loops.

ALGORITHM:

Take the number of rows as input from the user and store it in any variable.(‘r‘ in this case).

Run a loop ‘r’ number of times to iterate through each of the rows. From i=0 to i<r. The loop should be structured as for(i=0 ; i<r : i++).

Run a nested loop inside the main loop to print the spaces before the pyramid. From k=r to k>i +1. The loop should be structured as for(k=r; k>i+1 ;k–).

Inside this nested loop print white space.

Run another nested loop inside the main loop after the previous loop to print the stars in each column of a row. From j=0 to j<=i*2. The loop should be structured as for(j=0 ; j<i*2; j++).

Inside this nested loop print star.

Move to the next line by printing a new line . printf(“/n”).

Code in C:

```
#include<stdio.h>
int main()
{
int i,j,k,r;     //declaring integer variables i,j,k for loops and r for number of rows
printf("Enter the number of rows :\n");     //Asking user for input
scanf("%d",&r);      //saving number of rows in variable r
for(i=0;i<r;i++)    //outer loop for number of rows
  {
   for(k=r;k>i+1;k--)     //nested loop for number of spaces
     {
       printf(" ");     //printing spaces
     }
   for(j=0;j<=i*2;j++)     //nested loop for printing stars
     {
       printf("*");      //printing stars
     }
  printf("\n");     //printing newline
  }
}
```

16. Print Hollow Pyramid Star Pattern

PRINTING PATTERN:

```
    *
   * *
  *   *
 *     *
*******
```

PREREQUISITE:

Basic knowledge of C language and use of loops.

ALGORITHM:

Take the number of rows as input from the user and store it in any variable.(‘r‘ in this case).

Run a loop ‘r’ number of times to iterate through each of the rows. From i=0 to i<r. The loop should be structured as for(i=0 ; i<r : i++).

Run a nested loop inside the main loop to print the spaces before the pyramid. From k=r to k>i +1. The loop should be structured as for(k=r; k>i+1 ;k–).

Inside this nested loop print white space.

Run another nested loop inside the main loop after the previous loop to print the stars in each column of a row. From j=0 to j<=i*2. The loop should be structured as for(j=0 ; j<i*2; j++).

Inside this nested loop print star under the condition if(i==r-1).

Under else use a nested if else statement to print the remaining stars if(j==0||j>=i*2) and else print white space

Inside main loop move to the next line by printing a new line . printf(“/n”).

Code in C:

```
#include<stdio.h>
int main()
{
int i,j,k,r;     //declaring integer variables i,j,k for loops and r for number of rows
printf("Enter the number of rows :\n");     //Asking user for input
scanf("%d",&r);     //saving number of rows in variable r
for(i=0;i<r;i++)     //outer loop for number of rows
  {
    for(k=r;k>i+1;k--)//nested loop for number of spaces
      {
        printf(" ");//printing spaces
      }
    for(j=0;j<=i*2;j++)//nested loop for printing stars
      {
        if(i==r-1) //condition to print last row
          {
            printf("*"); //printing stars in last row
          }
```

```
            else //else condition to print rest of the stars
              {
                if(j==0||j>=i*2) //condition to print stars
                  {
                    printf("*"); //printing stars
                  }
                else //else condition to print spaces
                  {
                    printf(" "); //printing spaces
                  }
              }
          }
        printf("\n"); //printing newline
      }
}
```

17. Numbered Square Priting

PRINTING PATTERN:

1111
1111
1111
1111

PREREQUISITE:

Basic knowledge of C language and loops.

ALGORITHM:

Take number of rows/columns as input from the user and save it in any variable ('r'in this case).

Run a loop 'r' number of times to iterate through the rows. From i=0 to i<r. The loop should be structured as for(i=0 ; i<r ; i++).

Run a nested loop inside the previous loop to iterate through the columns. From j=0 to j<r. The loop should be structured as for(j=0 ; j<r ; j++).

Inside the nested loop print '1' to print '1' in each column of a row.

Inside the main loop print a newline to move to the next line after a row.

CODE IN C:

```
#include<stdio.h>
int main()
{
int i,j,r;                          //declaring integer variables i,j for loops , r for number of rows
printf("Enter the number of rows/columns :\n");  //asking user for number of rows/columns
scanf("%d",&r);                       //taking input and saving in variable r
for(i=0;i<r;i++)                      //loop for number of rows
  {
    for(j=0;j<r;j++)                  //loop for number of columns
      {
        printf("1");                  //printing 1
      }
    printf("\n");                     //printing newline
  }
}
```

18. Print Right Diamond Number Pattern Type8

PRINTING PATTERN:

2
34
567
891011
891011
567
34
2

PREREQUISITE:

Basic knowledge of C language and use of loops.

ALGORITHM:

Take the number of rows as input from the user and store it in any variable.(‘r‘ in this case).

Run a loop ‘r’ number of times to iterate through each of the rows. From i=0 to i<r. The loop should be structured as for(i=0 ; i<r : i++).

Use an if condition to to print the top half of the pyramid. if (i<=r/2). Then intialise count1=count+1 and Then run a loop from j=0 to j<=i. The loop should be structured as for(j=0 ; j<=i ; j++)

Inside this loop increment count and print it.

Outside this loop print a newline.

Else initialise count=count1;

Run a different loop from j=i to j<r-i. The loop should be structured as for(j=i ; j<r-i ; j++).

Inside this loop print count and increment it

outside the loop modify the value of count1 as count1=count1-(r-i)+1

then print a newline

CODE IN C:

```
#include<stdio.h>
int main()
{
int i,j,r,count,count1;                    //declaring integer variables i,j for loops , r for number of rows and count for increment in value
count=1; //initialising count
printf("Enter the number of rows(even) :\n");  //Asking user for input
scanf("%d",&r);                            //taking number of rows and saving it in variable r
for(i=0;i<r;i++)                           // loop for number of rows
  {
    if(i<r/2)
     {
       count1=count+1;
```

```
      for(j=0;j<=i;j++)                  // loop for digits per each row
        {
          count++;                       //incrementing count
          printf("%d",count);              //printing digits
        }
      printf("\n");                     // printing newline after each row
    }
   else                                 //else condition for bottom half
    {
      count=count1;                      //copying value
      for(j=0;j<r-i;j++)                 //loop to print digits
        {
          printf("%d",count);            //printing digits
          count++;                       //incrementing count
        }
      count1=count1-(r-i)+1;               //copying value
      printf("\n");                     //printing newline
    }

  }

}
```

19. Print Number Star Square Pattern

PRINTING PATTERN:

1*2*3*4
5*6*7*8
9*10*11*12
13*14*15*16

PREREQUISITE:

Basic knowledge of C language and use of loops.

ALGORITHM:

Take the number of rows/columns as input from the user and store it in any variable.(‘l‘ in this case).

Run a loop ‘l’ number of times to iterate through each of the rows. From i=0 to i<l. The loop should be structured as for(i=0 ; i<l : i++).

Inside this loop run another nested loop to iterate through the columns. From j=0 to j<l. The loop should be structured as for(j=0 ; j<l ; j++).

increment count and the run an if condition if(j==l-1) .

which means if it is the last column then print only count

Else print a star after count

outside the loop print a newline

CODE IN C:

```
#include<stdio.h>
int main()
{
int i,j,l,count=0;                    //declaring integers i,j for loops and l for number of rows
printf("Enter the number of rows/columns\n");  //Asking user for input
scanf("%d",&l);                       //Taking the input for number of rows
for(int i=0;i<l;i++)                  //Outer loop for number of rows
  {
    for(int j=0;j<l;j++)              //Inner loop for number of columns in each row
     {
       count++;                       //incrementing count
       if(j==l-1)                     //running if statement to not print star after the last column of digits
         {
           printf("%d",count);        //printing count
         }
       else                           //else statement to print star after count
         {
           printf("%d*",count);       //printing star after count
         }
     }
   printf("\n");                      //Printing a new line after each row has been printed.
  }
}
```

20. Print Number Star Right Diamond Pattern

PRINTING PATTERN:

6*6*6*6
5*5*5
4*4
3
3
4*4
5*5*5
6*6*6*6

PREREQUISITE:

Basic knowledge of C language and use of loops.

ALGORITHM:

Take the number of rows as input from the user and store it in any variable.(‘r‘ in this case).

Divide the value of ‘r’ by 2 and replace it in r. And give this value to count and increase count by 2.

Run a loop ‘r’ number of times to iterate through each of the rows. From i=0 to i<r. The loop should be structured as for(i=0 ; i<r : i++).

Run a loop from j=r to j>i. The loop should be structured as for(j=r; j>i ; j–)

Run an if statement if(j!=r). If true the print star and count else only print count.

Then in the outer if statement decrement count. Then print a newline

Outside this loop increment count and run a loop from i=0 to i<r. The loop should be structured as for(i=0 ; i<r : i++).

Run a nested loop from j=0 to j<=i . The loop should be structured as for(j=0 ; j<=i ; j++). Inside the loop run an if statement if(j!=0) then print star and digit else print only digit.

Outside the loop increment count and print a newline.

CODE IN C:

```
#include<stdio.h>
int main()
{
int i,j,r,count;//declaring integer variables i,j for loops , r for number of rows
printf("Enter the number of rows :\n");//asking user for the number of rows;
scanf("%d",&r);//taking number of rows and saving in variable r
r=r/2;
count=r+2;
for(i=0;i<r;i++) //loop for number of rows
  {
    for(j=r;j>i;j--)//loop to print digit in every column of a row
      {
        if(j!=r)
```

```
        {
          printf("*%d",count);//printing digit
        }
       else
        {
          printf("%d",count);//printing digit
        }
     }
   count--;
   printf("\n");//printing newline
  }
count++; //intialising count =3
for(i=0;i<r;i++) //loop for number of rows
  {
   for(j=0;j<=i;j++) //loop to print digit in every column of a row
     {
       if(j!=0)
        {
          printf("*%d",count);//printing digit
        }
       else
        {
          printf("%d",count);//printing digit
        }
     }
   count++; //incrementing count
   printf("\n"); //printing newline
  }
}
```