A Micro Project Report

on

Problem Solving using C Language

Submitted by

Bella Gnana Charmika(23471A05BF)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPET
(AUTONOMOUS)

Accredited by NAAC with A+ Grade and NBA under Tier-1

NIRF rank in the band of 201-300 and is an ISO 9001:2015 certified Approved by AICTE, New Delhi, Permanently affiliated to JNTU Kakinada, Approved by AICTE, Accredited by NBA and accredited ’ A+’ grade by NAAC Narasaraopet-522601, Palnadu(Dt.), Andhra Pradesh, India

2024-2025

NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPET

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that Bella Gnana Charmika, Roll No: 23471A05BF, a Second Year Student of the Department of Computer Science and Engineering, has completed the Micro Project Satisfactorily in " Problem Solving using C Language" for the Academic Year 2024-2025..

Project Co-Ordinator
Mr. Shaik Rafi, M.Tech., (Ph.D).
Asst. Professor

HEAD OF THE DEPARTMENT
Dr. S. N. Tirumala Rao, M.Tech., Ph.D.
Professor

INDEX

PASCAL TRIANGLE

AIM:

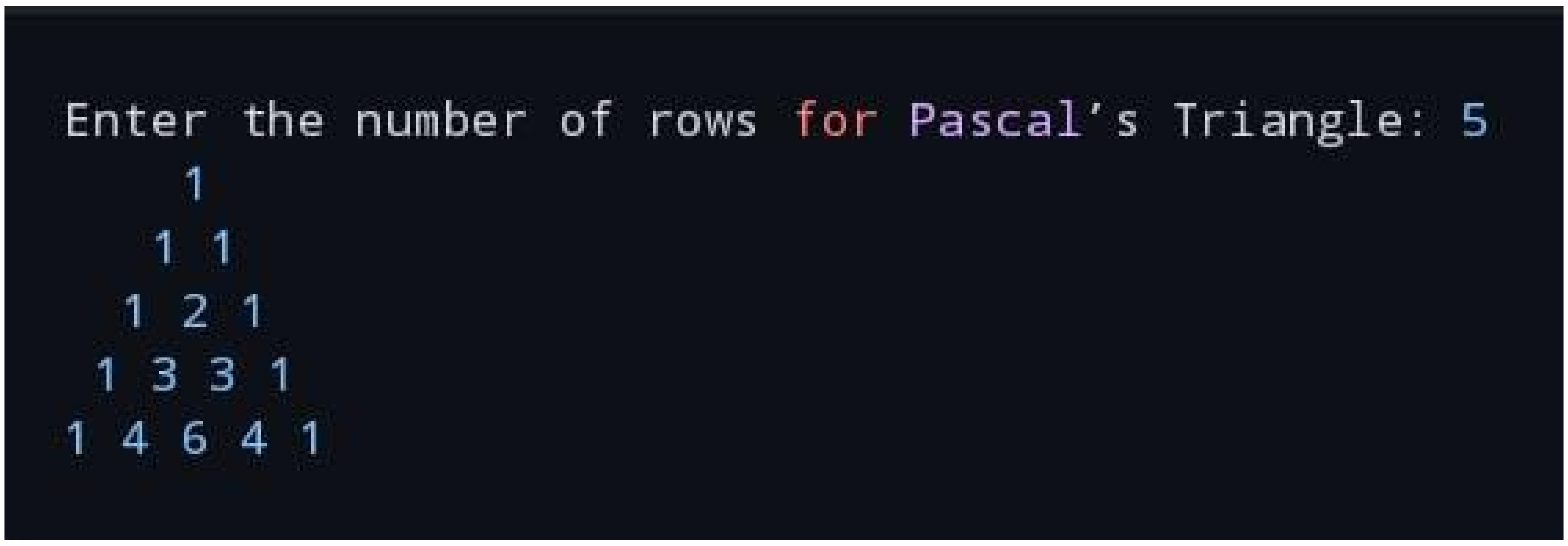
C Program to generate PASCAL Triangle

SOURCE CODE:

```
#include <stdio.h>
int factorial(int n)
{
   int fact = 1;
   for (int i = 1; i <= n; i++)
  {
      fact *= i;
   }
   return fact;
 }
int combination(int n, int r)
{
   return factorial(n) / (factorial(r) * factorial(n - r));
}
void printPascalTriangle(int rows)
{
   for (int i = 0; i < rows; i++)
  {
      for (int j = 0; j < rows - i - 1; j++)
      {
         printf(" ");
      }
      for (int j = 0; j <= i; j++)
      {
         printf("%d ", combination(i, j));
      }
      printf("\n");
   }
}
```

```
int main()
{
   int rows;
   printf("Enter the number of rows for Pascal’s Triangle: ");
   scanf("%d", &rows);
   printPascalTriangle(rows);
   return 0;
 }
```

OUTPUT:



```
Enter the number of rows for Pascal’s Triangle: 5
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
```

PLUS PATTERN USING STAR

AIM:

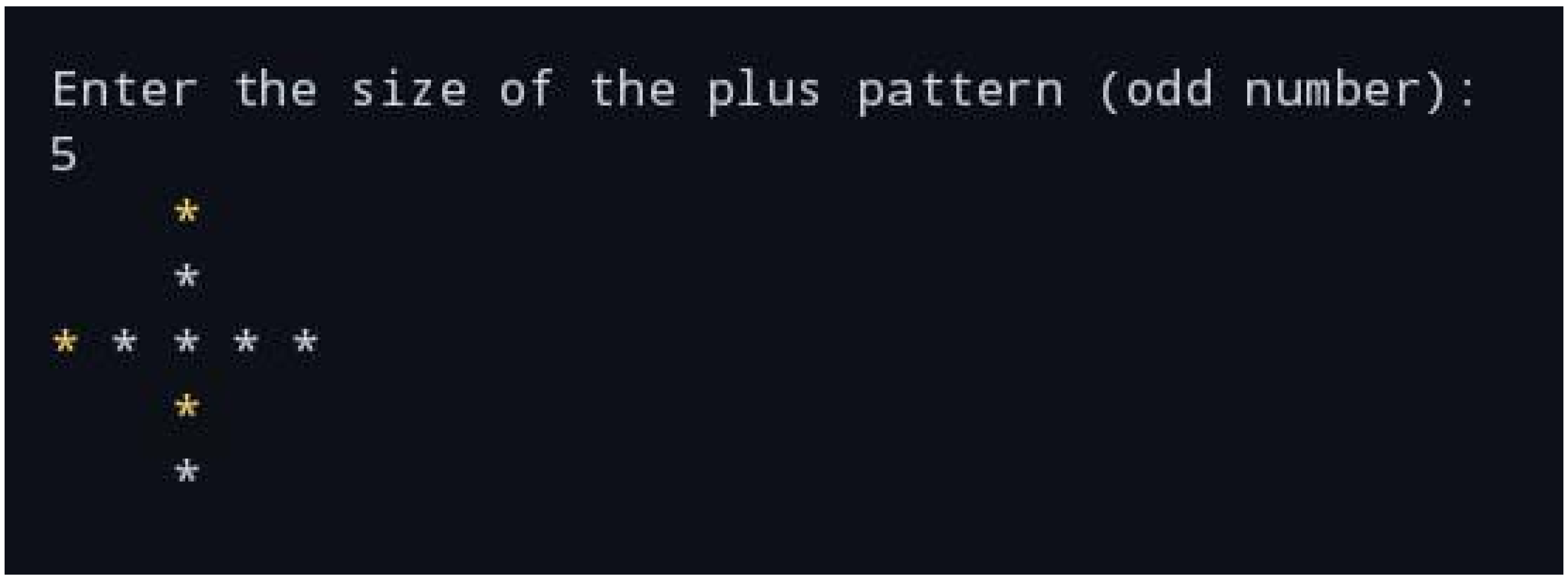
Generate Plus Pattern Using Star

SOURCE CODE:

```
#include <stdio.h>
void plus_pattern(int size)
{
    for (int i = 0; i < size; i++)
     {
         for (int j = 0; j < size; j++)
         {
               if (i == size / 2 || j == size / 2)
               {
                    printf("* ");
                }
                else
                {
                   printf("  ");
                 }
         }
            printf("\n");
     }
 }

int main()
{
    int size;
    printf("Enter the size of the plus pattern (odd number): ");
    scanf("%d", &size);
    if (size % 2 == 0)
   {
        printf("Please enter an odd number.\n");
        return 1;
    }
    plus_pattern(size);
    return 0;
 }
```

OUTPUT:



```
Enter the size of the plus pattern (odd number):
5
    *
    *
* * * * *
    *
    *
```

EQUILATERAL TRIANGLE SHAPE PATTERN

AIM:

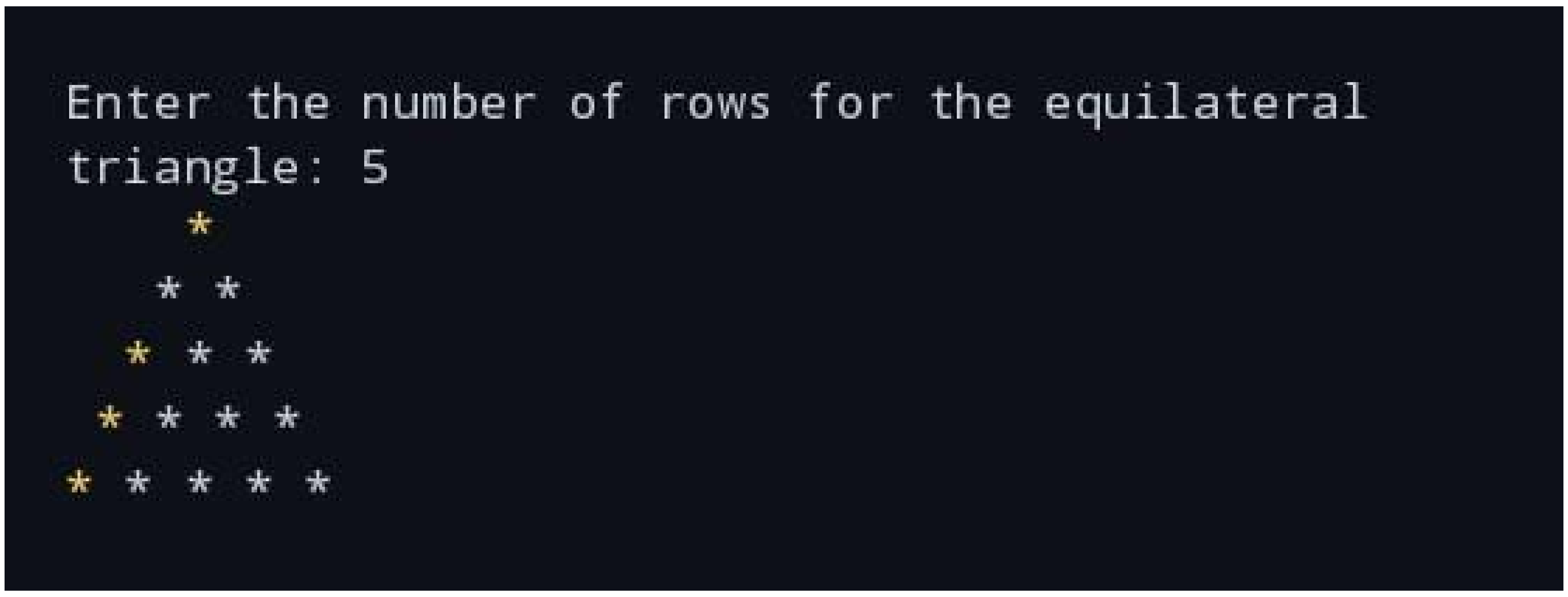
C Program to Generate Equilateral Triangle Shape Pattern

SOURCE CODE:

```
#include <stdio.h>
void printEquilateralTriangle(int rows)
{
  for (int i = 0; i < rows; i++)
  {
       for (int j = 0; j < rows - i - 1; j++)
      {
          printf(" ");
       }
       for (int j = 0; j <= i; j++)
      {
          printf("* ");
       }

         printf("\n");
   }
}
int main()
{
    int rows;
    printf("Enter the number of rows for the equilateral triangle: ");
    scanf("%d", &rows);
    printEquilateralTriangle(rows);
    return 0;
}
```

OUTPUT:



```
Enter the number of rows for the equilateral
triangle: 5
    *
   * *
  * * *
 * * * *
* * * * *
```

HOLLOW DIAMOND PATTERN

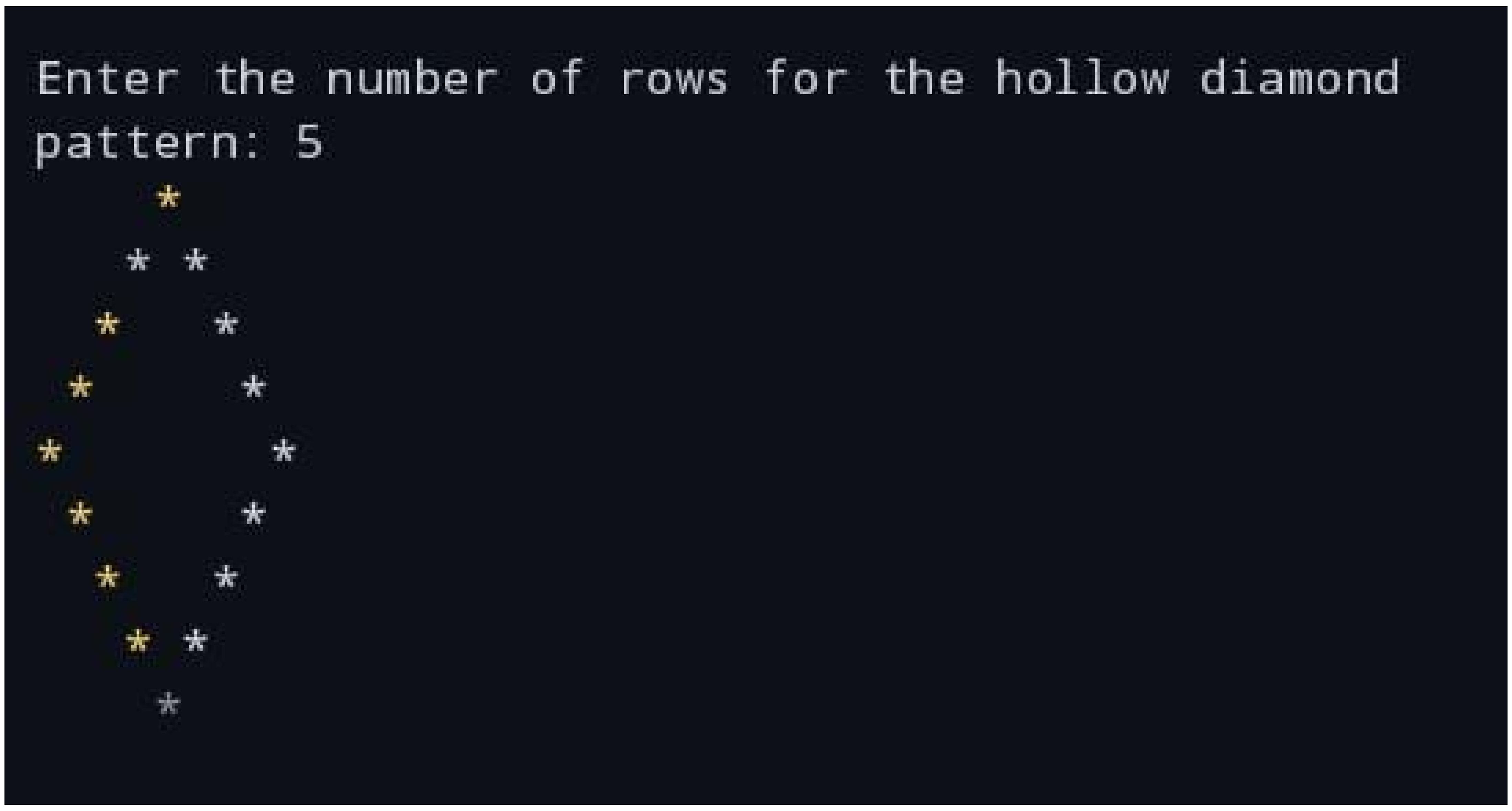
AIM:

C Program to Generate Hollow Diamond Pattern Using Stars

SOURCE CODE:

```
#include <stdio.h>
void printHollowDiamond(int n)
 {
    for (int i = 0; i < n; i++)
   {
        for (int j = 0; j < n - i - 1; j++)
       {
            printf(" ");
        }
        for (int j = 0; j < 2 * i + 1; j++)
        {
            if (j == 0 || j == 2 * i)
           {
                printf("*");
            }
            else
           {
                printf(" ");
            }
        }
        printf("\n");
    }
    for (int i = n - 2; i >= 0; i–)
    {
        for (int j = 0; j < n - i - 1; j++)
       {
            printf(" ");
        }
        for (int j = 0; j < 2 * i + 1; j++)
        {
            if (j == 0 || j == 2 * i)
            {
                printf("*");
            }
           else
           {
                printf(" ");
            }
        }
        printf("\n");
    }
int main()
{
    int n;
    printf("Enter the number of rows for the hollow diamond pattern: ");
    scanf("%d", &n);
    printHollowDiamond(n);
    return 0;
}
```

OUTPUT:



```
Enter the number of rows for the hollow diamond
pattern: 5
    *
   * *
  *   *
 *     *
*       *
 *     *
  *   *
   * *
    *
```

ALPHABET PATTERN

AIM:

Write a program to produce the following output:

```
A B C D E F G F E D C B A

A B C D E F   F E D C B A

A B C D E       E D C B A

A B C D           D C B A

A B C               C B A

A B                   B A

A                       A
```

SOURCE CODE:

```
#include <stdio.h>
void printPattern(int n)
{
    for (int i = 0; i < n; i++)
    {
        for (char ch = 'A'; ch < 'A' + n - i; ch++)
       {
            printf("%c ", ch);
        }
        for (int j = 0; j < 4 * i - 1; j++)
       {
            printf(" ");
        }
        if (i != 0)
        {
            for (char ch = 'A' + n - i - 1; ch >= 'A'; ch--)
           {
                printf("%c ", ch);
            }
        }
       else
       {
           for (char ch = 'A' + n - i - 2; ch >= 'A'; ch--) {
              printf("%c ", ch);
        }
       }
            printf("\n");
   }
}

int main()
{
   int n = 7; // Number of letters in the pattern (A-G)
   printPattern(n);
   return 0;
}
```

O/P:

```
A B C D E F G F E D C B A

A B C D E F   F E D C B A

A B C D E       E D C B A

A B C D           D C B A

A B C               C B A

A B                   B A

A                       A
```