NAME: ASHWEENA KHURRAM
CMS ID: 2754-2024
PROGRAMMING FUNDAMENTALS
LAB TASK#09
Q1.Creating a heart shape using nested loops in C involves carefully calculating the rows,
columns, and spaces to align stars (*) into the desired pattern.
Here's a simple example:

main.c

```c
#include <stdio.h>
#include <stdlib.h>

int main() {
    int n = 6; // Height of the heart

    // Upper part of the heart
    for (int i = n / 2; i <= n; i += 2) {
        // Print leading spaces
        for (int j = 1; j < n - i; j += 2) {
            printf(" ");    int main::n
        }
        // Print the first half of the stars
        for (int j = 1; j <= i; j++) {
            printf("*");
        }
        // Print the spaces between the two halves
        for (int j = 1; j <= n - i; j++) {
            printf(" ");
        }
        // Print the second half of the stars
```

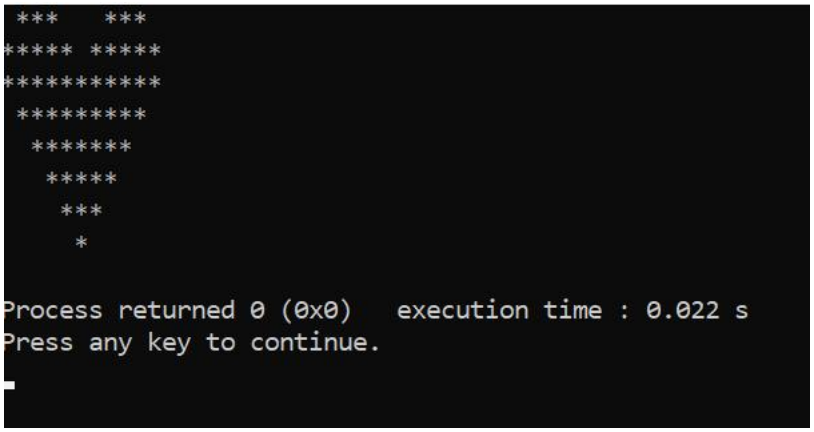
main.c

```c
        for (int j = 1; j <= i; j++) {
            printf("*");
        }
        printf("\n");
    }

    // Lower part of the heart
    for (int i = n; i >= 1; i--) {
        // Print leading spaces
        for (int j = 0; j < n - i; j++) {
            printf(" ");
        }
        // Print the stars for the row
        for (int j = 1; j <= (2 * i) - 1; j++) {
            printf("*");
        }
        printf("\n");
    }

    return 0;
}
```

OUTPUT



Q2.Diamond Pattern with Numbers

Write a program to generate a diamond pattern of numbers, where the user specifies

the number of rows for the top half. For example, if the user enters 4, the output

should be:

main.c

```c
#include <stdio.h>

int main() {
    int rows;

    // Ask the user for the number of rows for the top half
    printf("Enter the number of rows for the top half of the diamond:
    scanf("%d", &rows);

    // Top half of the diamond
    for (int i = 1; i <= rows; i++) {
        // Print spaces for alignment
        for (int j = 1; j <= rows - i; j++) {
            printf(" ");
        }
        // Print numbers in ascending order
        for (int j = 1; j <= i; j++) {
            printf("%d", j);
        }
        // Print numbers in descending order
        for (int j = i - 1; j >= 1; j--) {
```

main.c

```c
    for (int i = rows - 1; i >= 1; i--) {
        // Print spaces for alignment
        for (int j = 1; j <= rows - i; j++) {
            printf(" ");
        }
        // Print numbers in ascending order
        for (int j = 1; j <= i; j++) {
            printf("%d", j);
        }
        // Print numbers in descending order
        for (int j = i - 1; j >= 1; j--) {
            printf("%d", j);
        }
        printf("\n");
    }

    return 0;
}

```

OUTPUT

```
Enter the number of rows for the top half of the diamond: 4
   1
  121
 12321
1234321
 12321
  121
   1

Process returned 0 (0x0)   execution time : 3.554 s
Press any key to continue.
```

Q3. Right-Angled Triangle with Alphabet

Write a program to print a right-angled triangle where each row starts with the letter A and
continues alphabetically, wrapping around after Z. The user will input the number of rows.

For example, for 5 rows:

main.c

```c
#include <stdio.h>

int main() {
    int rows;
    char letter = 'A'; // Start with 'A'

    // Get the number of rows from the user
    printf("Enter the number of rows for the triangle: ");
    scanf("%d", &rows);

    // Outer loop for rows
    for (int i = 1; i <= rows; i++) {
        // Inner loop for printing letters
        for (int j = 1; j <= i; j++) {
            printf("%c ", letter);
            letter++; // Move to the next letter

            // Wrap around after 'Z'
            if (letter > 'Z') {
                letter = 'A';
            }
```

main.c

```c
    // Outer loop for rows
    for (int i = 1; i <= rows; i++) {
        // Inner loop for printing letters
        for (int j = 1; j <= i; j++) {
            printf("%c ", letter);
            letter++; // Move to the next letter

            // Wrap around after 'Z'
            if (letter > 'Z') {
                letter = 'A';
            }
        }
        printf("\n"); // Move to the next row
    }

    return 0;
}
```

OUTPUT

```
Enter the number of rows for the triangle: 8
A
B C
D E F
G H I J
K L M N O
P Q R S T U
V W X Y Z A B
C D E F G H I J

Process returned 0 (0x0)   execution time : 2.863 s
Press any key to continue.
```