

Assignment -4

While Loop:

1. Print Natural Numbers:

Write a program to print the first 10 natural numbers using a while loop.

Sol: #include <stdio.h>

```
int main() {  
    int i = 1;  
    printf("The first 10 natural numbers are:\n");  
    while (i <= 10) {  
        printf("%d\n", i);  
        i++;  
    }  
    return 0;  
}
```

O/p: The first 10 natural numbers are:

1
2
3
4
5
6
7
8

9

10

2. Sum of Digits:

Write a program to calculate the sum of the digits of a given integer using a while loop.

Sol: #include <stdio.h>

```
int main() {  
    int num, sum = 0, digit;  
    printf("Enter an integer: ");  
    scanf("%d", &num);  
    if (num < 0) {  
        num = -num;  
    }  
    while (num > 0) {  
        digit = num % 10;  
        sum += digit;  
        num /= 10;  
    }  
    printf("Sum of the digits: %d\n", sum);  
    return 0;  
}
```

o/p: Enter an integer: 234516

Sum of the digits: 21

3. Factorial of a Number:

Write a program to compute the factorial of a number using a while loop.

Sol: #include <stdio.h>

```
int main() {
    int num, factorial = 1, i;
    printf("Enter a non-negative integer: ");
    scanf("%d", &num);
    if (num < 0) {
        printf("Factorial is not defined for negative numbers.\n");
        return 1;
    }
    i = 1;
    while (i <= num) {
        factorial *= i;
        i++;
    }
    printf("Factorial of %d is %d\n", num, factorial);
    return 0;
}
```

O/p: Enter a non-negative integer: 5

Factorial of 5 is 120

4. Reverse a Number:

Write a program to reverse a given number using a while loop.

Sol: #include <stdio.h>

```
int main() {  
    int num, reversed = 0, digit;  
    printf("Enter an integer: ");  
    scanf("%d", &num);  
    while (num != 0) {  
        digit = num % 10;  
        reversed = reversed * 10 + digit;  
        num /= 10;  
    }  
    printf("Reversed number: %d\n", reversed);  
    return 0;  
}
```

o/p:

Enter an integer: 23456

Reversed number: 65432

5. Count Digits in a Number:

Write a program to count the number of digits in an integer using a while loop.

Sol: #include <stdio.h>

```
int main() {  
    int num, count = 0;  
    printf("Enter an integer: ");  
    scanf("%d", &num);
```

```
if (num == 0) {
    count = 1;
} else {
    if (num < 0) {
        num = -num;
    }
    while (num > 0) {
        num /= 10;
        count++;
    }
}

printf("Number of digits: %d\n", count);
return 0;
}
```

O/p: Enter an integer: 764311

Number of digits: 6

6. Print Multiplication Table:

Write a program to print the multiplication table of a given number using a while loop.

Sol: #include <stdio.h>

```
int main() {
    int num, i = 1;

    printf("Enter a number to print its multiplication table: ");
    scanf("%d", &num);
```

```
    printf("Multiplication table of %d:\n", num);  
while (i <= 10) {  
    printf("%d x %d = %d\n", num, i, num * i); // Print the table row  
    i++;  
}  
return 0;  
}
```

O/p: Enter a number to print its multiplication table: 3

Multiplication table of 3:

3 x 1 = 3

3 x 2 = 6

3 x 3 = 9

3 x 4 = 12

3 x 5 = 15

3 x 6 = 18

3 x 7 = 21

3 x 8 = 24

3 x 9 = 27

3 x 10 = 30

7. Check Palindrome Number:

Write a program to check if a number is a palindrome using a while loop.

Sol: #include <stdio.h>

```
int main() {
```

```
int num, original, reversed = 0, digit;
printf("Enter an integer: ");
scanf("%d", &num);
original = num;
if (num < 0) {
    printf("Negative numbers cannot be palindromes.\n");
    return 1;
}
while (num > 0) {
    digit = num % 10;
    reversed = reversed * 10 + digit;
    num /= 10;
}
if (original == reversed) {
    printf("%d is a palindrome.\n", original);
} else {
    printf("%d is not a palindrome.\n", original);
}
return 0;
}
```

O/p: Enter an integer: 121

121 is a palindrome.

Enter an integer: 123

123 is not a palindrome.

8. **Print Odd Numbers:**

Write a program to print all odd numbers between 1 and 50 using a while loop.

Sol: #include <stdio.h>

```
int main() {  
    int num = 1;  
    printf("Odd numbers between 1 and 50 are:\n");  
    while (num <= 50) {  
        printf("%d\n", num);  
        num += 2;  
    }  
    return 0;  
}
```

O/p: Odd numbers between 1 and 50 are:

1

3

5

7

9

11

13

15

17

19

21

23

25

27

29

31

33

35

37

39

41

43

45

47

49

9. Sum of Series:

Write a program to calculate the sum of the series:

$$S=1+2+3+\dots+n$$

using a while loop.

Sol: #include <stdio.h>

```
int main() {
```

```
int num, sum = 0, i = 1;
```

```

printf("Enter a positive integer: ");
scanf("%d", &num);
if (num <= 0) {
    printf("Please enter a positive integer.\n");
    return 1;
}
while (i <= num) {
    sum += i;
    i++;
}
printf("Sum of the series 1 + 2 + 3 + ... + %d is: %d\n", num, sum);
return 0;
}

```

O/p:

Enter a positive integer: 3

Sum of the series 1 + 2 + 3 + ... + 3 is: 6

Enter a positive integer: 5

Sum of the series 1 + 2 + 3 + ... + 5 is: 15

10. Find GCD of Two Numbers:

Write a program to compute the GCD of two numbers using a while loop.

Sol: #include <stdio.h>

```

int main() {
    int num1, num2, gcd;

```

```

printf("Enter two numbers: ");
scanf("%d %d", &num1, &num2);
int min = (num1 < num2) ? num1 : num2;
while (min > 0) {
    if (num1 % min == 0 && num2 % min == 0) {
        gcd = min;
        break;
    }
    min--;
}
printf("GCD of %d and %d is: %d\n", num1, num2, gcd);
return 0;
}

```

O/p: Enter two numbers: 24 2

GCD of 24 and 2 is: 2

For Loops:

1. Print Even Numbers:

Write a program to print all even numbers between 1 and 100 using a for loop.

Sol: #include <stdio.h>

```

int main() {
    for (int i = 1; i <= 100; i++) {
        if (i % 2 == 0) {

```

```

        printf("%d ", i);
    }
}
return 0;
}

```

O/p: 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46
 48 50 52 54 56 58 60 62 64 66 68 70 72 74 76 78 80 82 84 86 88 90 92
 94 96 98 100

2. Sum of First N Natural Numbers:

Write a program to calculate the sum of the first nnn natural numbers using a for loop.

Sol: #include <stdio.h>

```

    int main() {
        int n, sum = 0;
        printf("Enter the value of n: ");
        scanf("%d", &n);
        for (int i = 1; i <= n; i++) {
            sum += i; // Add i to sum
        }
        printf("Sum of first %d natural numbers is: %d\n", n, sum);

        return 0;
    }

```

O/p: Enter the value of n: 5

Sum of first 5 natural numbers is: 15

3. Factorial of a Number:

Write a program to calculate the factorial of a given number using a for loop.

Sol: #include <stdio.h>

```
int main() {  
    int n, factorial = 1;  
    printf("Enter a number: ");  
    scanf("%d", &n);  
    if (n < 0) {  
        printf("Factorial is not defined for negative numbers.\n");  
    } else {  
        for (int i = 1; i <= n; i++) {  
            factorial *= i;  
        }  
  
        printf("Factorial of %d is: %d\n", n, factorial);  
    }  
    return 0;  
}
```

O/p: Enter a number: 7

Factorial of 7 is: 5040

4. Fibonacci Series:

Write a program to generate the first nnn terms of the Fibonacci series using a for loop.

Sol: #include <stdio.h>

```
int main() {
    int n, first = 0, second = 1, next;
    printf("Enter the number of terms: ");
    scanf("%d", &n);
    if (n <= 0) {
        printf("Please enter a positive integer.\n");
    } else {
        printf("Fibonacci Series: ");
        if (n == 1) {
            printf("%d ", first);
        } else {
            printf("%d %d ", first, second);
            for (int i = 3; i <= n; i++) {
                next = first + second;
                printf("%d ", next);
                first = second;
                second = next;
            }
        }
        printf("\n");
    }
```

```
    }  
  
    return 0;  
}
```

O/p: Enter the number of terms: 6

Fibonacci Series: 0 1 1 2 3 5

5. Prime Number Check:

Write a program to check if a given number is prime using a for loop.

Sol: #include <stdio.h>

```
int main() {  
    int num, isPrime = 1;  
    printf("Enter a number: ");  
    scanf("%d", &num);  
    if (num <= 1) {  
        isPrime = 0;  
    } else {  
        for (int i = 2; i * i <= num; i++) {  
            if (num % i == 0) {  
                isPrime = 0;  
                break;  
            }  
        }  
    }  
}
```

```
if (isPrime) {  
    printf("%d is a prime number.\n", num);  
} else {  
    printf("%d is not a prime number.\n", num);  
}
```

```
return 0;
```

```
}
```

O/p: Enter a number: 3

3 is a prime number.

Enter a number: 10

10 is not a prime number.

6. Pattern Printing:

Print the following pattern using a for loop:

*

**

.....

Sol: #include <stdio.h>

```
int main() {
```

```
    int i, j;
```

```
    for (i = 1; i <= 4; i++) {
```



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

```
return 0;
```

```
}
```

O/p:

*

**

7. Sum of Squares of Numbers:

Write a program to calculate the sum of squares of the first *nnn* natural numbers using a for loop.

Sol: #include <stdio.h>

```
int main() {
```

```
int n, sum = 0;
```

```
printf("Enter the value of n: ");
```

```
scanf("%d", &n);
```

```
for (int i = 1; i <= n; i++) {  
    sum += i * i;  
}  
  
printf("Sum of squares of the first %d natural numbers is: %d\n",  
n, sum);  
  
return 0;  
}
```

O/p: Enter the value of n: 5

Sum of squares of the first 5 natural numbers is: 55

8. Power of a Number:

Write a program to compute (x raised to the power y) using a for loop.

Sol: #include <stdio.h>

```
int main() {  
    int x, y;  
    long long result = 1;  
    printf("Enter the base (x): ");  
    scanf("%d", &x);  
    printf("Enter the exponent (y): ");
```

```
scanf("%d", &y);

for (int i = 1; i <= y; i++) {

    result *= x;

}

printf("%d raised to the power %d is: %lld\n", x, y, result);

return 0;

}
```

O/p: Enter the base (x): 3

Enter the exponent (y): 2

3 raised to the power 2 is: 9

9. Reverse Counting:

Write a program to print numbers from 100 to 1 in reverse order using a for loop.

Sol: #include <stdio.h>

```
int main() {

    for (int i = 100; i >= 1; i--) {

        printf("%d ", i);

    }

    printf("\n");

    return 0;

}
```

O/p: 100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81
80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 64 63 62 61 60 59
58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37
36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15
14 13 12 11 10 9 8 7 6 5 4 3 2 1

10. Count Divisors of a Number:

Write a program to count the divisors of a given number using a for loop.

Sol: #include <stdio.h>

```
int main() {  
  
    int num, count = 0;  
  
    printf("Enter a number: ");  
  
    scanf("%d", &num);  
  
    for (int i = 1; i <= num; i++) {  
  
        if (num % i == 0) {  
  
            count++;  
  
        }  
  
    }  
  
    printf("The number of divisors of %d is: %d\n", num, count);  
  
    return 0;  
  
}
```

O/p: Enter a number: 12

The number of divisors of 12 is: 6

Do-While Loop:

1. Menu-Driven Calculator:

Write a menu-driven calculator using a do-while loop. Continue asking for user input until they choose to exit.

Sol #include <stdio.h>

```
int main() {
```

```
    int choice, num1, num2;
```

```
    do {
```

```
        printf("\n1. Add\n2. Subtract\n3. Multiply\n4. Divide\n5.  
Exit\n");
```

```
        printf("Choose an option: ");
```

```
        scanf("%d", &choice);
```

```
        if (choice == 5) break;
```

```
        printf("Enter two numbers: ");
```

```
        scanf("%d %d", &num1, &num2);
```

```
        if (choice == 1) printf("Result: %d\n", num1 + num2);
```

```
        else if (choice == 2) printf("Result: %d\n", num1 - num2);
```

```
    else if (choice == 3) printf("Result: %d\n", num1 * num2);  
    else if (choice == 4) {  
        if (num2 != 0) printf("Result: %.2f\n", (float)num1 / num2);  
        else printf("Cannot divide by 0!\n");  
    }  
} while (choice != 5);  
  
return 0;  
  
}
```

O/p:

1. Add
2. Subtract
3. Multiply
4. Divide
5. Exit

Choose an option: 1

Enter two numbers: 2 4

Result: 6

1. Add
2. Subtract
3. Multiply
4. Divide
5. Exit

Choose an option:

2. Print Numbers Until Zero:

Write a program to keep accepting numbers from the user and print them until the user enters zero.

Sol: #include <stdio.h>

```
int main() {  
  
    int number;  
  
    printf("Enter numbers (0 to stop):\n");  
  
    do {  
  
        scanf("%d", &number);  
  
        if (number != 0) {  
  
            printf("You entered: %d\n", number);  
  
        }  
  
    } while (number != 0);  
  
    printf("Program terminated.\n");  
  
    return 0;
```

```
}
```

O/p: Enter numbers (0 to stop):

234

You entered: 234

0

Program terminated.

3. **Validate Password:**

Write a program that asks for a password until the user provides the correct one using a do-while loop.

Sol: #include <stdio.h>

```
#include <string.h>
```

```
int main() {
```

```
char password[20];
```

```
const char correctPassword[] = "secure123";
```

```
printf("Enter the password: ");
```

```
do {
```

```
scanf("%19s", password);
```

```
if (strcmp(password, correctPassword) == 0) {
```

```
    printf("Access granted.\n");
```

```
    break;
```

```
} else {
```



```
        printf("Incorrect password. Try again: ");  
    }  
  
    } while (1);  
  
    return 0;  
}
```

O/p: Enter the password: likitha

Access granted.

Enter the password: pooja

Incorrect password. Try again:

4. Sum of Positive Numbers:

Write a program to read integers from the user and compute their sum. Stop when the user enters a negative number.

Sol: #include <stdio.h>

```
int main() {  
    int number;  
    int sum = 0;  
    printf("Enter numbers to sum (negative number to stop):\n");  
    do {  
        scanf("%d", &number);
```

```
        if (number >= 0) {  
            sum += number;  
        }  
  
    } while (number >= 0);  
  
    printf("The sum of positive numbers is: %d\n", sum);  
return 0;  
  
}
```

O/p: Enter numbers to sum (negative number to stop):

5

4

3

2

1

75

7

-8

The sum of positive numbers is: 97

5. Repeat Multiplication Table:

Write a program to repeatedly display the multiplication table of a number until the user decides to stop.

Sol: #include <stdio.h>

```
int main() {  
    int number, choice;  
  
    do {  
        printf("Enter a number to display its multiplication table: ");  
        scanf("%d", &number);  
        printf("Multiplication Table of %d:\n", number);  
        for (int i = 1; i <= 10; i++) {  
            printf("%d x %d = %d\n", number, i, number * i);  
        }  
        printf("\nDo you want to display another table? (1 for Yes, 0 for  
No): ");  
        scanf("%d", &choice);  
    } while (choice == 1);  
    printf("Program terminated.\n");  
  
    return 0;  
}
```

O/p: Enter a number to display its multiplication table: 5

Multiplication Table of 5:

$$5 \times 1 = 5$$

$$5 \times 2 = 10$$

$$5 \times 3 = 15$$

$$5 \times 4 = 20$$

$$5 \times 5 = 25$$

$$5 \times 6 = 30$$

$$5 \times 7 = 35$$

$$5 \times 8 = 40$$

$$5 \times 9 = 45$$

$$5 \times 10 = 50$$

Do you want to display another table? (1 for Yes, 0 for No): 0

Program terminated.

6. Guess the Number Game:

Write a program where the user guesses a predefined number. Continue the game until the correct number is guessed.

Sol: #include <stdio.h>

```
int main() {
```

```
    const int targetNumber = 42;
```

```
    int guess;
```

```
printf("Guess the number (between 1 and 100):\n");  
do {  
    printf("Enter your guess: ");  
    scanf("%d", &guess);  
    if (guess < targetNumber) {  
        printf("Too low. Try again.\n");  
    } else if (guess > targetNumber) {  
        printf("Too high. Try again.\n");  
    } else {  
        printf("Congratulations! You guessed the correct number:  
%d\n", targetNumber);  
    }  
  
    } while (guess != targetNumber);  
  
    return 0;  
}
```

O/p: Guess the number (between 1 and 100):

Enter your guess: 42

Congratulations! You guessed the correct number: 42

Guess the number (between 1 and 100):

Enter your guess: 34

Too low. Try again.

Enter your guess:

7. Input Validation:

Write a program to ensure that the user enters a number between 1 and 10. Prompt until a valid number is provided.

Sol: #include <stdio.h>

```
    int main() {  
  
        int number;  
  
        do {  
  
            printf("Enter a number between 1 and 10: ");  
  
            scanf("%d", &number);  
  
            if (number < 1 || number > 10) {  
  
                printf("Invalid input. Please try again.\n");  
  
            }  
  
        } while (number < 1 || number > 10);  
  
        printf("You entered a valid number: %d\n", number);  
  
        return 0;  
  
    }
```

O/p: Enter a number between 1 and 10: 6

You entered a valid number: 6

Enter a number between 1 and 10: 65

Invalid input. Please try again.

8. Calculate Average:

Write a program to calculate the average of a series of numbers entered by the user. Stop when the user enters zero.

Sol: #include <stdio.h>

```
int main() {
```

```
    int number, count = 0;
```

```
    double sum = 0, average;
```

```
    printf("Enter numbers to calculate the average (0 to stop):\n");
```

```
    do {
```

```
        scanf("%d", &number);
```

```
        if (number != 0) {
```

```
            sum += number;
```

```
            count++;
```

```
        }
```

```
    } while (number != 0);
```

```
if (count > 0) {  
    average = sum / count;  
    printf("The average of the entered numbers is: %.2lf\n", average);  
} else {  
    printf("No numbers were entered.\n");  
}  
  
return 0;  
}
```

O/p: Enter numbers to calculate the average (0 to stop):

3

4

3

2

0

The average of the entered numbers is: 3.00

9. Print Alphabets:

Write a program to print lowercase alphabets from 'a' to 'z' using a do-while loop.

Sol: #include <stdio.h>


```
int main() {  
    char letter = 'a';  
    printf("Lowercase alphabets from 'a' to 'z':\n");  
    do {  
        printf("%c ", letter);  
        letter++;  
    } while (letter <= 'z');  
    printf("\n");  
  
    return 0;  
}
```

O/p:

Lowercase alphabets from 'a' to 'z':

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

10. Count Digits of a Number:

Write a program to count the number of digits in a number entered by the user using a do-while loop

Sol: #include <stdio.h>

```
int main() {  
    int number, count = 0;
```

```
printf("Enter a number: ");  
scanf("%d", &number);  
if (number == 0) {  
    count = 1;  
} else {  
  
    do {  
        number /= 10;  
        count++;  
    } while (number != 0);  
printf("The number of digits is: %d\n", count);  
  
return 0;  
}  
}
```

O/p: Enter a number: 24672

The number of digits is: 5

Problem statements with respect to Pattern printing using For as well as while Loop

1. Pascal's Triangle

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

Sol: Using for Loop:

```
#include <stdio.h>

void pascal_triangle_for(int n) {
    int i, j, val;
    for(i = 0; i < n; i++) {
        for(j = 0; j < n - i - 1; j++) {
            printf(" ");
        }
        val = 1;
        for(j = 0; j <= i; j++) {
            printf("%d ", val);
            val = val * (i - j) / (j + 1);
        }
    }
}
```

```
        printf("\n");  
    }  
}
```

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

Using while Loop:

```
#include <stdio.h>  
  
void pascal_triangle_while(int n) {  
    int i = 0, j, val;
```

```
while (i < n) {  
  
    j = 0;  
    while (j < n - i - 1) {  
        printf(" ");  
        j++;  
    }  
    val = 1;  
    j = 0;  
    while (j <= i) {  
        printf("%d ", val);  
        val = val * (i - j) / (j + 1);  
        j++;  
    }  
    printf("\n");  
    i++;  
}  
  
int main() {  
    int n;
```

```
printf("Enter the number of rows for Pascal's Triangle: ");  
scanf("%d", &n);  
pascal_triangle_while(n);  
return 0;  
}
```

O/p: 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
```

2. Binary Pattern

```
1  
01  
101  
0101  
10101
```

Sol: Using for Loop:

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

```
int main() {  
    int rows = 5;  
    for (int i = 1; i <= rows; i++) {  
        for (int j = 1; j <= i; j++) {  
            printf("%d", j % 2);  
        }  
        printf("\n");  
    }  
    return 0;  
}
```

Using while Loop:

```
#include <stdio.h>  
  
int main() {  
    int rows = 5, i = 1, j;  
    while (i <= rows) {  
        j = 1;  
        while (j <= i) {  
            printf("%d", j % 2);  
            j++;  
        }  
    }
```

```
        printf("\n");  
        i++;  
    }  
    return 0;  
}
```

O/p:

```
1  
10  
101  
1010  
10101
```

3. Floyd's Triangle (Numbers)

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

Sol: Using for Loop:

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



```
int main() {  
    int rows = 5, num = 1;  
    for (int i = 1; i <= rows; i++) {  
        for (int j = 1; j <= i; j++) {  
            printf("%d ", num++);  
        }  
        printf("\n");  
    }  
    return 0;  
}
```

Using while Loop:

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

```
int main() {  
    int rows = 5, num = 1, i = 1, j;  
    while (i <= rows) {  
        j = 1;  
        while (j <= i) {  
            printf("%d ", num++);  
            j++;  
        }  
    }
```

```
        printf("\n");  
        i++;  
    }  
    return 0;  
}
```

O/p:

1

2 3

4 5 6

7 8 9 10

11 12 13 14 15

4. Inverted Right-Angled Triangle (Numbers)

12345

1234

123

12

1

Sol: Using for Loop:

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

```
int main() {  
    int rows = 5;  
    for (int i = rows; i >= 1; i--) {  
        for (int j = 1; j <= i; j++) {  
            printf("%d", j);  
        }  
        printf("\n");  
    }  
    return 0;  
}
```

Using while Loop:

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

```
int main() {  
    int rows = 5, i = rows, j;  
    while (i >= 1) {  
        j = 1;  
        while (j <= i) {  
            printf("%d", j);  
            j++;  
        }  
    }
```

```
        printf("\n");
        i--;
    }
    return 0;
}
```

O/p:

12345

1234

123

12

1

5. Diamond (Stars)

```

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

*

Sol: Using For loop

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

```
int main() {
```

```
    int rows = 5, i, j, space;
```

```
    printf("Diamond (Stars) using for loop:\n");
```

```
    for(i = 1; i <= rows; i++) {
```

```
        for(j = 1; j <= rows - i; j++) {
```

```
            printf(" ");
```

```
        }
```

```
        for(j = 1; j <= (2 * i - 1); j++) {
```

```
            printf("*");
```

```
        }
```

```
        printf("\n");
```

```
    }
```

```
    for(i = rows - 1; i >= 1; i--) {
```

```
        for(j = 1; j <= rows - i; j++) {
```

```
            printf(" ");
```

```
        }
```

```

        for(j = 1; j <= (2 * i - 1); j++) {
            printf("*");
        }
        printf("\n");
    }
    return 0;
}

#include <stdio.h>

int main() {
    int rows = 5, i, j, space;
    printf("Diamond (Stars) using while loop:\n");
    i = 1;
    while(i <= rows) {
        j = 1;
        while(j <= rows - i) {
            printf(" ");
            j++;
        }
        j = 1;

```

```
    while(j <= (2 * i - 1)) {  
        printf("*");  
        j++;  
    }  
    printf("\n");  
    i++;  
}  
i = rows - 1;  
while(i >= 1) {  
    j = 1;  
    while(j <= rows - i) {  
        printf(" ");  
        j++;  
    }  
    j = 1;  
    while(j <= (2 * i - 1)) {  
        printf("*");  
        j++;  
    }  
    printf("\n");
```

```
        i--;  
    }  
    return 0;  
}
```

O/p: Diamond (Stars) using while loop:

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

6. Inverted Pyramid (Stars)

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


*

Sol: Using for Loop:

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

```
void inverted_pyramid_for(int n) {
```

```
    int i, j;
```

```
    for (i = 0; i < n; i++) {
```

```
        for (j = 0; j < i; j++) {
```

```
            printf(" ");
```

```
        }
```

```
        for (j = 0; j < (2 * (n - i) - 1); j++) {
```

```
            printf("*");
```

```
        }
```

```
        printf("\n");
```

```
    }
```

```
}
```

```
int main() {
```

```
    int n;
```

```
    printf("Enter the number of rows for the inverted pyramid: ");
```

```
    scanf("%d", &n);
```

```
    inverted_pyramid_for(n);  
    return 0;  
}  
}
```

Using while Loop:

```
#include <stdio.h>  
  
void inverted_pyramid_while(int n) {  
    int i = 0, j;  
    while (i < n) {  
        j = 0;  
        while (j < i) {  
            printf(" ");  
            j++;  
        }  
        j = 0;  
        while (j < (2 * (n - i) - 1)) {  
            printf("*");  
            j++;  
        }  
    }
```

```

        printf("\n");

        i++;
    }
}

int main() {
    int n;

    printf("Enter the number of rows for the inverted pyramid: ");
    scanf("%d", &n);

    inverted_pyramid_while(n);

    return 0;
}

```

O/p: Enter the number of rows for the inverted pyramid: 5

```
*****
```

```
*****
```

```
*****
```

```
***
```

```
*
```