

While Loop

1. Print Natural Numbers:

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

```
#include <stdio.h>

int main()

{ int i = 1;

  while (i <= 10){

    printf("%d ", i);

    i++;

  }

  return 0;

}
```

Output:

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.

```
#include <stdio.h>

int main() {

  int num, sum = 0;

  printf("Enter a number: ");

  scanf("%d", &num);

  while (num != 0) {

    sum += num % 10;

    num /= 10;

  }

  printf("Sum of digits: %d\n", sum);

  return 0;

}
```

Output:

Enter a number: 23

Sum of digits: 5

3. Factorial of a Number:

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

```
#include <stdio.h>

int main() {
    int num, fact = 1;

    printf("Enter a number: ");

    scanf("%d", &num);

    while (num > 1) {
        fact *= num;
        num--;
    }

    printf("Factorial: %d\n", fact);

    return 0;
}
```

Output:

Enter a number: 6

Factorial: 720

4. Reverse a Number:

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

```
#include <stdio.h>

int main() {
    int num, reverse = 0;

    printf("Enter a number: ");

    scanf("%d", &num);

    while (num != 0) {
        reverse = reverse * 10 + num % 10;
        num /= 10;
    }

    printf("Reversed number: %d\n", reverse);

    return 0;
}
```

Output:

Enter a number: 34

Reversed number: 43

5. **Count Digits in a Number:**

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

```
#include <stdio.h>

int main() {
    int num, count = 0;
    printf("Enter a number: ");
    scanf("%d", &num);
    while (num != 0) {
        count++;
        num /= 10;
    }
    printf("Number of digits: %d\n", count);
    return 0;
}
```

Output:

Enter a number: 12

Number of digits: 2

6. **Print Multiplication Table:**

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

```
#include <stdio.h>

int main() {
    int num, i = 1;
    printf("Enter a number: ");
    scanf("%d", &num);
    while (i <= 10) {
        printf("%d x %d = %d\n", num, i, num * i);
        i++;
    }
    return 0;
}
```

Output:

Enter a number: 8

8 x 1 = 8

8 x 2 = 16

8 x 3 = 24

8 x 4 = 32

8 x 5 = 40

8 x 6 = 48

8 x 7 = 56

8 x 8 = 64

8 x 9 = 72

8 x 10 = 80

7. Check Palindrome Number:

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

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

```
int main() {
```

```
    int num, original, reverse = 0, remainder;
```

```
    printf("Enter a number: ");
```

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

```
    original = num;
```

```
    while (num != 0) {
```

```
        remainder = num % 10;
```

```
        reverse = reverse * 10 + remainder;
```

```
        num /= 10;
```

```
    }
```

```
    if (original == reverse) {
```

```
        printf("The number is a palindrome.\n");
```

```
    } else {
```

```
        printf("The number is not a palindrome.\n");
```

```
    }
```

```
    return 0;
```

```
}
```

Output:

Enter a number: 565

The number is a palindrome.

8.Print Odd Numbers:

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

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

```
int main() {
```

```
    int i = 1;
```

```
    while (i <= 50) {
```

```
        if (i % 2 != 0) {
```

```
            printf("%d ", i);
```

```
        }
```

```
        i++;
```

```
    }
```

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

```
    return 0;
```

```
}
```

Output:

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:

$S=1+2+3+...+n$ using a while loop.

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

```
int main() {
```

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

```
    printf("Enter a number: ");
```

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

```
    while (i <= n) {
```

```
        sum += i;
```

```
        i++;
```

```
    }
```

```
printf("Sum of series: %d\n", sum);  
return 0;  
}
```

Output:

Enter a number: 55

Sum of series: 1540

10. Find GCD of Two Numbers:

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

```
#include <stdio.h>  
  
int main() {  
    int num1, num2;  
    printf("Enter two numbers: ");  
    scanf("%d %d", &num1, &num2);  
    while (num1 != num2) {  
        if (num1 > num2) {  
            num1 -= num2;  
        } else {  
            num2 -= num1;  
        }  
    }  
    printf("GCD: %d\n", num1);  
    return 0;  
}
```

Output:

Enter two numbers: 22 55

GCD: 11

For Loop

1. Print Even Numbers:

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

```
#include <stdio.h>

int main() {
    for (int i = 2; i <= 100; i += 2) {
        printf("%d ", i);
    }
    return 0;
}
```

Output:

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.

```
#include <stdio.h>

int main() {
    int n, sum = 0;
    printf("Enter a number: ");
    scanf("%d", &n);
    for (int i = 1; i <= n; i++) {
        sum += i;
    }
    printf("Sum of first %d natural numbers is: %d", n, sum);
    return 0;
}
```

Output:

Enter a number: 3

Sum of first 3 natural numbers is: 6

3. Factorial of a Number:

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

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

```

int main() {
    int n, factorial = 1;
    printf("Enter a number: ");
    scanf("%d", &n);
    for (int i = 1; i <= n; i++) {
        factorial *= i;
    }
    printf("Factorial of %d is: %d", n, factorial);
    return 0;
}

```

Output:

Enter a number: 23

Factorial of 23 is: 862453760

4. **Fibonacci Series:**

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

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

```

int main() {
    int n;
    printf("Enter the number of terms: ");
    scanf("%d", &n);
    int a = 0, b = 1, next;
    printf("Fibonacci Series: ");
    for (int i = 1; i <= n; i++) {
        if (i == 1) {
            printf("%d ", a);
            continue;
        }
        if (i == 2) {
            printf("%d ", b);
            continue;
        }
    }
}

```



```

    next = a + b;

    a = b;

    b = next;

    printf("%d ", next);
}

return 0;
}

```

Output:

Enter the number of terms: 22

Fibonacci Series: 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946

5. Prime Number Check:

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

```

#include <stdio.h>

int main() {

    int num, isPrime = 1;

    printf("Enter a number: ");

    scanf("%d", &num);

    if (num <= 1) {

        isPrime = 0;

    }

    for (int i = 2; i <= num / 2; i++) {

        if (num % i == 0) {

            isPrime = 0;

            break;

        }

    }

    if (isPrime) {

        printf("%d is a prime number.", num);

    } else {

        printf("%d is not a prime number.", num);

    }

}

```

```
    return 0;
}
```

Output:

Enter a number: 7

7 is a prime number.

6. **Pattern Printing:**

Print the following pattern using a for loop:

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

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

```
int main() {
    int n = 4; // Number of rows

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

    return 0;
}
```

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.

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

```
int main() {
    int n, sum = 0;

    printf("Enter a number: ");

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

```

for (int i = 1; i <= n; i++) {
    sum += i * i;
}

printf("Sum of squares of first %d natural numbers is: %d", n, sum);

return 0;
}

```

Output:

Enter a number: 12

Sum of squares of first 12 natural numbers is: 650

8. **Power of a Number:**

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

```

#include <stdio.h>

int main() {
    int x, y, result = 1;

    printf("Enter base (x): ");
    scanf("%d", &x);

    printf("Enter exponent (y): ");
    scanf("%d", &y);

    for (int i = 1; i <= y; i++) {
        result *= x;
    }

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

    return 0;
}

```

Output:

Enter base (x): 23

Enter exponent (y): 12

23 raised to the power 12 is: 480779105

9. **Reverse Counting:**

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

```

#include <stdio.h>

```

```

int main() {
    for (int i = 100; i >= 1; i--) {
        printf("%d ", i);
    }
    return 0;
}

```

Output:

```

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.

```

#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", num, count);
    return 0;
}

```

Output:

Enter a number: 11

The number of divisors of 11 is: 2

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.

```
#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;

}
```

Output:

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

Choose an option: 3

Enter two numbers: 22 3

Result: 66

2. **Print Numbers Until Zero:**

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

```
#include <stdio.h>

int main() {

    int num;

    do {

        printf("Enter a number: ");

        scanf("%d", &num);

        if (num != 0)

            printf("You entered: %d\n", num);

    } while(num != 0);

    return 0;

}
```

Output:

Enter a number: 33

You entered: 33

Enter a number: 22

You entered: 22

3. **Validate Password:**

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

```
#include <stdio.h>

int main() {

    char password[20];

    do {

        printf("Enter password: ");

        scanf("%s", password);

        if (strcmp(password, "1234") != 0) {

            printf("Incorrect password. Try again.\n");

        }

    } while (strcmp(password, "1234") != 0);

}
```

```

    }
} while (strcmp(password, "1234") != 0);
printf("Password correct!\n");
return 0;
}

```

Output:

Enter password: 1234

Password correct!

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.

```

#include <stdio.h>

int main() {
    int num, sum = 0;
    do {
        printf("Enter a number: ");
        scanf("%d", &num);
        if (num >= 0) {
            sum += num;
        }
    } while(num >= 0);
    printf("Sum of positive numbers: %d\n", sum);
    return 0;
}

```

Output:

Enter a number: 12

Enter a number: 232

Enter a number: 33

Enter a number: 0

Enter a number: -12

Sum of positive numbers: 277

5. Repeat Multiplication Table:

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

```
#include <stdio.h>

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

Output:

Enter a number to display its multiplication table: 5

5 x 1 = 5

5 x 2 = 10

5 x 3 = 15

5 x 4 = 20

5 x 5 = 25

5 x 6 = 30

5 x 7 = 35

5 x 8 = 40

5 x 9 = 45

5 x 10 = 50

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

Goodbye!

6. Guess the Number Game:

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

```
#include <stdio.h>

int main() {

    int number = 42;

    int guess;

    do {

        printf("Guess the number (between 1 and 100): ");

        scanf("%d", &guess);

        if (guess < number) {

            printf("Too low! Try again.\n");

        } else if (guess > number) {

            printf("Too high! Try again.\n");

        }

    } while (guess != number);

    printf("Congratulations! You've guessed the correct number.\n");

    return 0;

}
```

Output:

Guess the number (between 1 and 100): 34

Too low! Try again.

Guess the number (between 1 and 100): 56

Too high! Try again.

Guess the number (between 1 and 100): 42

Congratulations! You've guessed the correct number.

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.

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

```

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

```

Output:

```

Enter a number between 1 and 10: 44
Invalid input. Please enter a number between 1 and 10.
Enter a number between 1 and 10: 22
Invalid input. Please enter a number between 1 and 10.
Enter a number between 1 and 10: 2
You entered: 2

```

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.

```

#include <stdio.h>

int main() {
    int num, sum = 0, count = 0;
    do {
        printf("Enter a number : ");
        scanf("%d", &num);
        if (num != 0) {
            sum += num;
            count++;
        }
    }
}

```

```

    } while(num != 0);
    if (count != 0) {
        printf("Average: %.2f\n", (float)sum / count);
    } else {
        printf("No numbers were entered.\n");
    }
    return 0;
}

```

Output:

Enter a number: 9

Enter a number: 5

Enter a number: 0

Average: 7.00

9. **Print Alphabets:**

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

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

```

int main() {
    char ch = 'a';
    do {
        printf("%c ", ch);
        ch++;
    } while (ch <= 'z');
    printf("\n");
    return 0;
}

```

Output:

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.

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

```

int main() {

```

```

int num, count = 0;
printf("Enter a number: ");
scanf("%d", &num);
do {
    num /= 10;
    count++;
} while (num != 0);
printf("Number of digits: %d\n", count);
return 0;
}

```

Output:

Enter a number: 678

Number of digits: 3

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

```

Using For loop

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

```
int factorial(int n) {
```

```
    if (n == 0 || n == 1)
```

```
        return 1;
```

```
    return n * factorial(n - 1);
```

```
}
```

```
int combination(int n, int r) {
```

```

        return factorial(n) / (factorial(r) * factorial(n - r));
    }

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

```

Using while loop

```

#include <stdio.h>

int factorial(int n) {
    if (n == 0 || n == 1)
        return 1;
    return n * factorial(n - 1);
}

int combination(int n, int r) {
    return factorial(n) / (factorial(r) * factorial(n - r));
}

int main() {
    int rows = 5;
    int i = 0;
    while (i < rows) {
        int space = 1;
        while (space <= rows - i) {

```

```

        printf(" ");
        space++;
    }
    int j = 0;
    while (j <= i) {
        printf("%d ", combination(i, j));
        j++;
    }
    printf("\n");
    i++;
}

return 0;
}

```

2. Binary Pattern

1
 01
 101
 0101
 10101

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;
```

```
}
```

3. Floyd's Triangle (Numbers)

1

2 3

4 5 6

7 8 9 10

11 12 13 14 15

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;
}

```

4. Inverted Right-Angled Triangle (Numbers)

12345

1234

123

12

1

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;
}

```

5. Diamond (Stars)

```

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

```

*

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++) {
```

```
        // Print leading spaces
```

```
        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;
```

```
}
```

Using while loop

```
#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;
}

```

6. Inverted Pyramid (Stars)

```

*****

*****

*****

***

*

```

Using For loop

```

#include <stdio.h>

int main() {
    int rows = 5;
    for (int i = rows; i >= 1; i--) {
        for (int space = 0; space < rows - i; space++) {
            printf(" ");
        }
        for (int j = 1; j <= 2 * i - 1; j++) {
            printf("*");
        }
        printf("\n");
    }
    return 0;
}

```

Using while loop

```

#include <stdio.h>

int main() {
    int rows = 5;

```

```
int i = rows;
while (i >= 1) {
    int space = 0;
    while (space < rows - i) {
        printf(" ");
        space++;
    }
    int j = 1;
    while (j <= 2 * i - 1) {
        printf("*");
        j++;
    }
    printf("\n");
    i--;
}
return 0;
}
```