Contents

[LOOPING 3](#_Toc159179926)

[while Loop: 3](#_Toc159179927)

[do...while Loop: 3](#_Toc159179928)

[for Loop: 4](#_Toc159179929)

[foreach Loop: 5](#_Toc159179930)

[break and continue Statements: 5](#_Toc159179931)

[PROGRAMS 7](#_Toc159179932)

[While loop : 7](#_Toc159179933)

[Program to Print Numbers from 1 to 10 7](#_Toc159179934)

[Program to Calculate Factorial of a Number 7](#_Toc159179935)

[Program to Find Sum of Natural Numbers 8](#_Toc159179936)

[Program to Reverse a Number 9](#_Toc159179937)

[Program to Find GCD (Greatest Common Divisor) of Two Numbers 9](#_Toc159179938)

[Do while loop 10](#_Toc159179939)

[Program to Print Numbers from 1 to 10 10](#_Toc159179940)

[Program to Calculate Factorial of a Number 11](#_Toc159179941)

[Program to Print Multiplication Table 12](#_Toc159179942)

[Program to Calculate Average of Numbers 12](#_Toc159179943)

[Program to Generate Fibonacci Series 13](#_Toc159179944)

[Program to Implement a Simple Menu 14](#_Toc159179945)

[For loop 15](#_Toc159179946)

[Program to Calculate Factorial of a Number 15](#_Toc159179947)

[Program to Find Sum of Natural Numbers 16](#_Toc159179948)

[Program to Print Multiplication Table 17](#_Toc159179949)

[Program to Generate Fibonacci Series 17](#_Toc159179950)

[Program to Print Even Numbers from 1 to 20 18](#_Toc159179951)

[Program to Calculate Sum of Even Numbers from 1 to 100 18](#_Toc159179952)

[Program to Print ASCII Values of Characters A to Z 19](#_Toc159179953)

[Foreach loop 20](#_Toc159179954)

[Program to Iterate Over an Array 20](#_Toc159179955)

[Program to Iterate Over Characters in a String 20](#_Toc159179956)

[OTHER PROGRAM 21](#_Toc159179957)

[while loop to play a simple guessing game: 21](#_Toc159179958)

[Write a program that prints on the console the numbers from 1 to N, which are not divisible by 3 and 7 simultaneously. The number N should be read from the standard input. 22](#_Toc159179959)

[Write a program that reads from the console a series of integers and prints the smallest and largest of them. 23](#_Toc159179960)

[Write a program that prints all possible cards from a standard deck of cards, without jokers (there are 52 cards: 4 suits of 13 cards). 24](#_Toc159179961)

[Program to find sum of all digits of a given number 25](#_Toc159179962)

[Program to reverse digits of a given number 26](#_Toc159179963)

[Program to Check Whether the Entered Number is an Armstrong Number or Not 27](#_Toc159179964)

[Program to Print all the Armstrong Numbers from 1 to 1000 27](#_Toc159179965)

[Program to Display the Factors of the Entered Number 28](#_Toc159179966)

[Program to check whether a given number is Palindrome or not 29](#_Toc159179967)

# LOOPING

## while Loop:

-The while loop repeatedly executes a block of statements as long as a specified condition evaluates to true.

Syntax:

while (condition)

{

// Statements to be executed

}

Example:

int i = 0;

while (i < 5)

{

Console.WriteLine("Iteration: " + i);

i++;

}

## do...while Loop:

-The do...while loop is similar to the while loop, but it ensures that the block of statements is executed at least once, even if the condition is initially false.

Syntax:

do

{

// Statements to be executed

} while (condition);

Example:

int i = 0;

do

{

Console.WriteLine("Iteration: " + i);

i++;

} while (i < 5);

## for Loop:

-The for loop executes a block of statements a specified number of times. It consists of three parts: initialization, condition, and iteration.

- It's used when you know exactly how many times you want to repeat a block of code.

Syntax:

for (initialization; condition; iteration)

{

// Statements to be executed

}

Explanation:

-initialization: Here, you set up any variables or conditions needed before starting the loop. This part is executed only once at the beginning.

-condition: This is a condition that, if true, allows the loop to continue executing. If false, the loop stops.

-iteration: This is the action that happens after each iteration of the loop. It typically increments or decrements a variable.

Example:

for (int i = 0; i < 5; i++)

{

Console.WriteLine("Iteration: " + i);

}

## foreach Loop:

-The foreach loop iterates over elements in a collection, such as arrays or collections, and executes a block of statements for each element.

-It's particularly useful when you want to work with each item in a collection, such as an array or a list, without worrying too much about the details of managing indexes or counting elements.

Syntax:

foreach (type variable in collection)

{

// Statements to be executed

}

Explanation:

-type: The data type of each item in the collection.

-variable: The name of a variable that represents each item in the collection.

-collection: The collection of items you want to loop through.

Example:

int[] numbers = { 1, 2, 3, 4, 5 };

foreach (int number in numbers)

{

Console.WriteLine("Number: " + number);

}

## break and continue Statements:

-break: Terminates the loop immediately and transfers control to the statement following the loop.

-continue: Skips the remaining code inside the loop for the current iteration and proceeds to the next iteration.

Example:

int i = 0;

while (true)

{

if (i == 3)

{

break;

}

Console.WriteLine("Iteration: " + i);

i++;

}

Example:

for (int i = 0; i < 5; i++)

{

if (i == 2)

{

continue;

}

Console.WriteLine("Iteration: " + i);

}

# PROGRAMS

## While loop :

### Program to Print Numbers from 1 to 10

using System;

class PrintNumbers

{

static void Main()

{

int i = 1;

while (i <= 10)

{

Console.WriteLine(i);

i++;

}

}

}

### Program to Calculate Factorial of a Number

using System;

class Factorial

{

static void Main()

{

Console.Write("Enter a number: ");

int num = int.Parse(Console.ReadLine());

int factorial = 1;

int i = 1;

while (i <= num)

{

factorial \*= i;

i++;

}

Console.WriteLine("Factorial of {0} is: {1}", num, factorial);

}

}

### Program to Find Sum of Natural Numbers

using System;

class SumOfNumbers

{

static void Main()

{

Console.Write("Enter a positive integer: ");

int num = int.Parse(Console.ReadLine());

int sum = 0;

int i = 1;

while (i <= num)

{

sum += i;

i++;

}

Console.WriteLine("Sum of first {0} natural numbers is: {1}", num, sum);

}

}

### Program to Reverse a Number

using System;

class ReverseNumber

{

static void Main()

{

Console.Write("Enter a number: ");

int number = int.Parse(Console.ReadLine());

int reversedNumber = 0;

while (number > 0)

{

int digit = number % 10;

reversedNumber = reversedNumber \* 10 + digit;

number /= 10;

}

Console.WriteLine("Reversed number: " + reversedNumber);

}

}

### Program to Find GCD (Greatest Common Divisor) of Two Numbers

using System;

class GCD

{

static void Main()

{

Console.Write("Enter first number: ");

int num1 = int.Parse(Console.ReadLine());

Console.Write("Enter second number: ");

int num2 = int.Parse(Console.ReadLine());

int gcd = 0;

int smaller = (num1 < num2) ? num1 : num2;

int i = 1;

while (i <= smaller)

{

if (num1 % i == 0 && num2 % i == 0)

{

gcd = i;

}

i++;

}

Console.WriteLine("GCD of {0} and {1} is: {2}", num1, num2, gcd);

}

}

## Do while loop

### Program to Print Numbers from 1 to 10

using System;

class PrintNumbers

{

static void Main()

{

int i = 1;

do

{

Console.WriteLine(i);

i++;

} while (i <= 10);

}

}

### Program to Calculate Factorial of a Number

using System;

class Factorial

{

static void Main()

{

Console.Write("Enter a number: ");

int num = int.Parse(Console.ReadLine());

int factorial = 1;

int i = 1;

do

{

factorial \*= i;

i++;

} while (i <= num);

Console.WriteLine("Factorial of {0} is: {1}", num, factorial);

}

}

### Program to Print Multiplication Table

using System;

class MultiplicationTable

{

static void Main()

{

Console.Write("Enter the number for multiplication table: ");

int number = int.Parse(Console.ReadLine());

int i = 1;

Console.WriteLine("Multiplication table of {0}:", number);

do

{

Console.WriteLine("{0} x {1} = {2}", number, i, number \* i);

i++;

} while (i <= 10);

}

}

### Program to Calculate Average of Numbers

using System;

class AverageCalculator

{

static void Main()

{

int sum = 0;

int count = 0;

int number;

Console.WriteLine("Enter numbers (enter 0 to calculate average):");

do

{

number = int.Parse(Console.ReadLine());

sum += number;

count++;

} while (number != 0);

double average = (double)sum / (count - 1);

Console.WriteLine("Average of entered numbers is: " + average);

}

}

### Program to Generate Fibonacci Series

using System;

class FibonacciSeries

{

static void Main()

{

Console.Write("Enter the number of terms in Fibonacci series: ");

int n = int.Parse(Console.ReadLine());

int first = 0, second = 1;

int i = 0;

Console.WriteLine("Fibonacci series:");

do

{

Console.Write(first + " ");

int next = first + second;

first = second;

second = next;

i++;

} while (i < n);

}

}

### Program to Implement a Simple Menu

using System;

class SimpleMenu

{

static void Main()

{

int choice;

do

{

Console.WriteLine("1. Print Hello");

Console.WriteLine("2. Print Goodbye");

Console.WriteLine("3. Exit");

Console.Write("Enter your choice: ");

choice = int.Parse(Console.ReadLine());

switch (choice)

{

case 1:

Console.WriteLine("Hello!");

break;

case 2:

Console.WriteLine("Goodbye!");

break;

case 3:

Console.WriteLine("Exiting program.");

break;

default:

Console.WriteLine("Invalid choice. Please try again.");

break;

}

} while (choice != 3);

}

}

## For loop

### Program to Calculate Factorial of a Number

using System;

class Factorial

{

static void Main()

{

Console.Write("Enter a number: ");

int num = int.Parse(Console.ReadLine());

int factorial = 1;

for (int i = 1; i <= num; i++)

{

factorial \*= i;

}

Console.WriteLine("Factorial of {0} is: {1}", num, factorial);

}

}

## Program to Find Sum of Natural Numbers

using System;

class SumOfNumbers

{

static void Main()

{

Console.Write("Enter a positive integer: ");

int num = int.Parse(Console.ReadLine());

int sum = 0;

for (int i = 1; i <= num; i++)

{

sum += i;

}

Console.WriteLine("Sum of first {0} natural numbers is: {1}", num, sum);

}

}

## Program to Print Multiplication Table

using System;

class MultiplicationTable

{

static void Main()

{

Console.Write("Enter the number for multiplication table: ");

int number = int.Parse(Console.ReadLine());

Console.WriteLine("Multiplication table of {0}:", number);

for (int i = 1; i <= 10; i++)

{

Console.WriteLine("{0} x {1} = {2}", number, i, number \* i);

}

}

}

### Program to Generate Fibonacci Series

using System;

class FibonacciSeries

{

static void Main()

{

Console.Write("Enter the number of terms in Fibonacci series: ");

int n = int.Parse(Console.ReadLine());

int first = 0, second = 1;

Console.WriteLine("Fibonacci series:");

for (int i = 0; i < n; i++)

{

Console.Write(first + " ");

int next = first + second;

first = second;

second = next;

}

}

}

### Program to Print Even Numbers from 1 to 20

using System;

class PrintEvenNumbers

{

static void Main()

{

Console.WriteLine("Even numbers from 1 to 20:");

for (int i = 2; i <= 20; i += 2)

{

Console.WriteLine(i);

}

}

}

## Program to Calculate Sum of Even Numbers from 1 to 100

using System;

class SumOfEvenNumbers

{

static void Main()

{

int sum = 0;

for (int i = 2; i <= 100; i += 2)

{

sum += i;

}

Console.WriteLine("Sum of even numbers from 1 to 100: " + sum);

}

}

### Program to Print ASCII Values of Characters A to Z

using System;

class PrintASCIIValues

{

static void Main()

{

Console.WriteLine("ASCII values of characters A to Z:");

for (char c = 'A'; c <= 'Z'; c++)

{

Console.WriteLine("{0}: {1}", c, (int)c);

}

}

}

## Foreach loop

## Program to Iterate Over an Array

using System;

class ArrayIteration

{

static void Main()

{

int[] numbers = { 1, 2, 3, 4, 5 };

Console.WriteLine("Array elements:");

foreach (int num in numbers)

{

Console.WriteLine(num);

}

}

}

### Program to Iterate Over Characters in a String

using System;

class StringIteration

{

static void Main()

{

string message = "Hello, world!";

Console.WriteLine("Characters in the string:");

foreach (char ch in message)

{

Console.WriteLine(ch);

}

}

}

# OTHER PROGRAM

### while loop to play a simple guessing game:

using System;

class GuessingGame

{

static void Main()

{

Random random = new Random();

int secretNumber = random.Next(1, 101); // Generate a random number between 1 and 100

int attempts = 0;

int guess = 0;

Console.WriteLine("Welcome to the Guessing Game!");

Console.WriteLine("Try to guess the secret number between 1 and 100.");

while (guess != secretNumber)

{

Console.Write("Enter your guess: ");

guess = int.Parse(Console.ReadLine());

attempts++;

if (guess < secretNumber)

{

Console.WriteLine("Too low! Try again.");

}

else if (guess > secretNumber)

{

Console.WriteLine("Too high! Try again.");

}

}

Console.WriteLine("Congratulations! You guessed the secret number {0} in {1} attempts.", secretNumber, attempts);

}

}

### Write a program that prints on the console the numbers from 1 to N, which are not divisible by 3 and 7 simultaneously. The number N should be read from the standard input.

using System;

class Program

{

static void Main()

{

Console.Write("Enter a number (N): ");

int N = int.Parse(Console.ReadLine());

Console.WriteLine("Numbers from 1 to {0} not divisible by 3 and 7 simultaneously:", N);

for (int i = 1; i <= N; i++)

{

if (i % 3 != 0 || i % 7 != 0)

{

Console.WriteLine(i);

}

}

}

}

### Write a program that reads from the console a series of integers and prints the smallest and largest of them.

using System;

class Program

{

static void Main()

{

Console.WriteLine("Enter a series of integers (type 'done' to finish):");

int smallest = int.MaxValue;

int largest = int.MinValue;

string input = Console.ReadLine();

while (input != "done")

{

int number;

if (int.TryParse(input, out number))

{

smallest = Math.Min(smallest, number);

largest = Math.Max(largest, number);

}

else

{

Console.WriteLine("Invalid input. Please enter a valid integer or 'done' to finish.");

}

input = Console.ReadLine();

}

if (smallest != int.MaxValue && largest != int.MinValue)

{

Console.WriteLine("Smallest number: " + smallest);

Console.WriteLine("Largest number: " + largest);

}

else

{

Console.WriteLine("No valid integers were entered.");

}

}

}

- int.TryParse is a method provided by the System namespace in C#. It's used to attempt to convert a string representation of a number into an integer type.

- out is a C# keyword used to pass arguments to a method by reference.

- int.TryParse(input, out number) attempts to convert the input string (input) into an integer. If successful, it assigns the parsed integer value to the number variable and returns true. If unsuccessful, it returns false, indicating that the input is not a valid integer.

### Write a program that prints all possible cards from a standard deck of cards, without jokers (there are 52 cards: 4 suits of 13 cards).

using System;

class Program

{

static void Main()

{

string[] suits = { "Hearts", "Diamonds", "Clubs", "Spades" };

string[] ranks = { "2", "3", "4", "5", "6", "7", "8", "9", "10", "Jack", "Queen", "King", "Ace" };

Console.WriteLine("All possible cards from a standard deck of cards:");

foreach (string suit in suits)

{

foreach (string rank in ranks)

{

Console.WriteLine($"{rank} of {suit}");

}

}

}

}

### Program to find sum of all digits of a given number

using System;

namespace system {

class sumofdigits {

static void Main(String[] args) {

int a = 567, sum = 0, b;

//condition to check if the number is not 0

while (a != 0) {

b = a % 10; //extract a digit

sum = sum + b; //adding the digits

a = a / 10; //remained number

}

Console.WriteLine("The sum of the digits is: " + sum);

}

}

}

### Program to reverse digits of a given number

using System;

namespace system {

class reverse {

static void Main(String[] args) {

int a = 721, rev = 0, b;

//condition to check if the number is not 0

while (a != 0) {

b = a % 10; //extract a digit

rev = (rev \* 10) + b; //reverse the digits logic

a = a / 10; //remained number

}

Console.WriteLine("The reverse of the number is: " + rev);

}

}

}

### Program to Check Whether the Entered Number is an Armstrong Number or Not

using System;

namespace ConsoleApplication

{

class Program

{

static void Main(string[] args)

{

int number, remainder, sum = 0;

Console.Write("enter the Number");

number = int.Parse(Console.ReadLine());

for (int i = number; i > 0; i = i / 10)

{

remainder = i % 10;

sum = sum + remainder\*remainder\*remainder;

}

if (sum == number)

{

Console.Write("Entered Number is an Armstrong Number");

}

else

Console.Write("Entered Number is not an Armstrong Number");

Console.ReadLine();

}

}

}

### Program to Print all the Armstrong Numbers from 1 to 1000

using System;

class Program

{

static void Main()

{

int a, b, c, d;

for (int i = 1; i <= 1000; i++)

{

a = i / 100;

b = (i - a \* 100) / 10;

c = (i - a \* 100 - b \* 10);

d = a \* a \* a + b \* b \* b + c \* c \* c;

if (i == d)

{

System.Console.WriteLine("{0}", i);

}

}

Console.Read();

}

}

### Program to Display the Factors of the Entered Number

using System;

namespace Program

{

class Program

{

static void Main(string[] args)

{

int num, x;

Console.WriteLine("Enter the Number ");

num = int.Parse(Console.ReadLine());

Console.WriteLine("The Factors are : ");

for (x = 1; x <= num; x++)

{

if (num % x == 0)

{

Console.WriteLine(x);

}

}

Console.ReadLine();

}

}

}

### Program to check whether a given number is Palindrome or not

using System;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

int number = 0;

int tNumber = 0;

int rem = 0;

int rev = 0;

Console.Write("Enter Number : ");

tNumber = number = int.Parse(Console.ReadLine());

//To find out total number of digits in number

while (number > 0)

{

rem = number %10;

rev = rev \* 10 + rem;

number = number / 10;

}

if (rev == tNumber)

Console.WriteLine("Given Number is Palindrome");

else

Console.WriteLine("Given Number is not a Palindrome");

}

}

}