**20 C# Programs By**

**Surya Teja Chandolu**

**27/01/2022**

|  |
| --- |
| 1. C# Program: To Print MULTIPLICATION TABLE of given number |
| Code: |
| using System;  namespace MultiplicationTableWhileLoop  {  internal class Program  {  static void Main(string[] args)  {  //Initilize  int input;  //User Input  Console.Write("Enter any number: ");  input = Convert.ToInt32(Console.ReadLine());  //Logic and Output  int i = 1;  while(i <= 10)  {  Console.WriteLine(input + "x" + i + "=" + input \* i);  i++;  }    Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
| 1. C# Program: Print FACTORIAL of a given number |
| Code: |
| using System;  namespace FactorialOfANumber  {  internal class Program  {  static void Main(string[] args)  {  //Intialzing  int input, fact = 1;  //User Input  Console.Write("Enter any Number to find Factorial: ");  input = Convert.ToInt32(Console.ReadLine());  //Logic  for(int i = 1; i <= input; i++)  {  fact = fact \* i;  }  //Output  Console.WriteLine($"Factorial of {input} is {fact}");  Console.ReadLine();  }  }  } |
| Output |
|  |

|  |
| --- |
| 1. C# Program: Print SUM OF N Natural Numbers |
| Code: |
| using System;  namespace SumOfNNaturalNumbersForLoop  {  internal class Program  {  static void Main(string[] args)  {  //Intialzing  int input, sum = 0;  //User Input  Console.Write("Enter any number: ");  input = Convert.ToInt32(Console.ReadLine());  //Logic for sum of N natural numbers  for (int i = 0; i <= input; i++)  {  sum = sum + i;  }  //Output  Console.WriteLine($"Sum of {input} natural numbers are {sum}");  Console.ReadLine();  }  }  } |
| Output |
|  |

|  |
| --- |
| 1. C# Program: Print FACTORIAL using FUNCTION |
| Code: |
| using System;  namespace FactorialUsingFunctions  {  internal class Program  {  //Factorial Function  public static int Factorial(int input)  {  int fact = 1;  for (int i = 1; i <= input; i++)  fact = fact \* i;  return fact;  }  //Output Function  public static void Output(int input)  {  Console.WriteLine($"Factorial of {input} is {Factorial(input)}");  }  public static void Main(string[] args)  {  //Initilize  int num, num1;  //User Input  Console.Write("Enter number: ");  num = Convert.ToInt32(Console.ReadLine());  Console.Write("Enter number: ");  num1 = Convert.ToInt32(Console.ReadLine());  Output(num);  Output(num1);  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
| 1. C# Program: Print FACTORIAL using RECURSION |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace FactorialUsingFunctions  {  internal class Program  {  //Factorial Function Using Recursion  public static int Factorial(int input)  {  if (input == 0)  return 1;  else  return input \* Factorial(input - 1);  }  //Output Function  public static void Output(int input)  {  Console.WriteLine($"Factorial of {input} is {Factorial(input)}");  }  public static void Main(string[] args)  {  //Initilize  int num, num1;  //User Input  Console.Write("Enter number: ");  num = Convert.ToInt32(Console.ReadLine());  Console.Write("Enter number: ");  num1 = Convert.ToInt32(Console.ReadLine());  Output(num);  Output(num1);  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
| 1. C# Program: Print FACTORS of given number |
| Code: |
| using System;  namespace FactorsOfANumber  {  internal class Program  {  static void Main(string[] args)  {  //Intialzing  int input;  //User Input  Console.Write("Enter any Number to find Factors of a number: ");  input = Convert.ToInt32(Console.ReadLine());  //Logic and Output  Console.Write($"Factors of {input} are ");  for (int i = 1; i <= input; i++)  {  if(input%i == 0)  {  Console.Write($"{i} ");  }  }  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
| 1. C# Program: Print POWER of Given numbers [a power b] |
| Code: |
| using System;  namespace PowerOfGivenNumber  {  internal class Program  {  static void Main(string[] args)  {  int bas, power, value = 1;  Console.Write("Enter number for Base: ");  bas = Convert.ToInt32(Console.ReadLine());  Console.Write("Enter of a number Power: ");  power = Convert.ToInt32(Console.ReadLine());  for(int i = 1; i <= power; i++)  {  value = value \* bas;  }  Console.WriteLine($"{bas} to the power {power} is {value}");  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
| 1. C# Program: PRIME NUMBER or Not |
| Code: |
| using System;  namespace PrimeOrNot  {  internal class Program  {  static void Main(string[] args)  {  int input, i;  Console.Write("Enter number: ");  input = Convert.ToInt32(Console.ReadLine());  for(i = 2; i < input; i++)  {  if (input % 2 == 0)  break;  }  if (i == input)  Console.WriteLine($"{input} is a Prime number");  else  Console.WriteLine($"{input} is not a Prime number");  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
| 1. C# Program: PRIME NUMBER check [Using FUNCTION] |
| Code: |
| using System;  namespace PrimeOrNot  {  internal class Program  {  public static void Prime(int input)  {  int i;  for (i = 2; i < input; i++)  {  if (input % i == 0)  break;  }  if (i == input)  Console.WriteLine($"{input} is a Prime number");  else  Console.WriteLine($"{input} is not a Prime number");  }  static void Main(string[] args)  {  Console.Write("Enter number: ");  Prime(Convert.ToInt32(Console.ReadLine()));  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
| 1. C# Program: PRIME NUMBERS in RANGE |
| Code: |
| using System;  namespace PrimeOrNot  {  internal class Program  {  //Prime using Function  public static bool Prime(int input)  {  int i;  for (i = 2; i < input; i++)  {  if (input % i == 0)  break;  }  if (i == input)  return true;  else  return false;  }  static void Main(string[] args)  {  int start, end;  Console.Write("Enter starting number: ");  start = Convert.ToInt32(Console.ReadLine());  Console.Write("Enter ending number: ");  end = Convert.ToInt32(Console.ReadLine());  for(int i = start; i <= end; i++)  {  if(Prime(i))  Console.WriteLine(i);  }  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
| 1. C# Program: FIBONACCI SERIES |
| Code: |
| using System;  namespace FibonacciSeries  {  internal class Program  {  static void Main(string[] args)  {  int num, firstNumber = 0, secondNumber = 1, value;  Console.Write("Enter number: ");  num = Convert.ToInt32(Console.ReadLine());  Console.Write("Fibonacci Serice are 0 1 ");  for(int i = 1; i < num - 2; i++)  {  value = firstNumber + secondNumber;  firstNumber = secondNumber;  secondNumber = value;  Console.Write($"{value} ");  }  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
| 1. C# Program: ARMSTRONG NUMBER |
| Code: |
| using System;  namespace ArmstrongNumber  {  internal class Program  {  static void Main(string[] args)  {  int num, rem, sum = 0, temp;  Console.Write("Enter number: ");  num = Convert.ToInt32(Console.ReadLine());  temp = num;  while(num > 0)  {  rem = num % 10;  sum = sum + (rem\*rem\*rem);  num = num / 10;  }  if(temp == sum)  Console.WriteLine($"{temp} is a Armstrong number");  else  Console.WriteLine($"{temp} is not a Armstrong number");  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
| 1. C# Program: ARMSTRONG NUMBER [using FUNCTION] |
| Code: |
| using System;  namespace ArmstrongNumber  {  internal class Program  {  public static bool Armstrong(int num)  {  int rem, sum = 0, temp;  temp = num;  while (num > 0)  {  rem = num % 10;  sum = sum + (rem \* rem \* rem);  num = num / 10;  }  if (temp == sum)  return true;  else  return false;  }  static void Main(string[] args)  {  int num;  Console.Write("Enter number: ");  num = Convert.ToInt32(Console.ReadLine());  if (Armstrong(num) == true)  Console.WriteLine($"{num} is Armstrong number");  else  Console.WriteLine($"{num} is not a Armstrong number");  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
| 1. C# Program: ARMSTRONG NUMBERS IN RANGE |
| Code: |
| using System;  namespace ArmstrongNumber  {  internal class Program  {  public static bool Armstrong(int num)  {  int rem, sum = 0, temp;  temp = num;  while (num > 0)  {  rem = num % 10;  sum = sum + (rem \* rem \* rem);  num = num / 10;  }  if (temp == sum)  return true;  else  return false;  }  static void Main(string[] args)  {  int num, num1;  Console.Write("Enter first number: ");  num = Convert.ToInt32(Console.ReadLine());  Console.Write("Enter Second number: ");  num1 = Convert.ToInt32(Console.ReadLine());  for(int i = num; i <= num1; i++)  {  if(Armstrong(i))  Console.WriteLine($"Armstrong number are {i}");  }  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
| 1. C# Program: SUM OF DIGITS of given number |
| Code: |
| using System;  namespace SumOfDigits  {  internal class Program  {  static void Main(string[] args)  {  int num, rem, sum = 0, temp;  Console.Write("Enter number: ");  num = Convert.ToInt32(Console.ReadLine());  temp = num;  while (num > 0)  {  rem = num % 10;  num = num / 10;  sum = sum + rem;  }  Console.WriteLine($"Sum of digites of {temp} is {sum}");  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
| 1. C# Program: REVERSE OF given number |
| Code: |
| using System;  namespace ReverseNumber  {  internal class Program  {  static void Main(string[] args)  {  int num, smp, rev = 0, rem;  Console.Write("Enter a number: ");  num = Convert.ToInt32(Console.ReadLine());  smp = num;  while(num > 0)  {  rem = num % 10;  rev = (rev \* 10) + rem;  num = num / 10;  }  Console.WriteLine($"Reverse number for {smp} is {rev}");  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
| 1. C# Program: PALINDROME NUMBER |
| Code: |
| using System;  namespace Palindrome  {  internal class Program  {  static void Main(string[] args)  {  int num , temp, sum = 0, rem;  Console.Write("Enter a number: ");  num = Convert.ToInt32(Console.ReadLine());  temp = num;  while (num > 0)  {  rem = num % 10;  sum = (sum \* 10) + rem;  num = num / 10;  }  if(temp == sum)  Console.WriteLine($"{temp} is a Palindrome number");  else  Console.WriteLine($"{temp} is not a Palindrome number");  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
| 1. C# Program: SWAP NUMBERS using THIRD VARIABLE |
| Code: |
| using System;  namespace SwapNumber  {  internal class Program  {  static void Main(string[] args)  {  int firstNumber, secondNumber, temp;  Console.Write("Enter First number: ");  firstNumber = Convert.ToInt32(Console.ReadLine());  Console.Write("Enter Second number: ");  secondNumber = Convert.ToInt32(Console.ReadLine());  Console.WriteLine($"Numbers before Swap are First number {firstNumber} and Second number {secondNumber}");  temp = firstNumber;  firstNumber = secondNumber;  secondNumber = temp;  Console.WriteLine($"Numbers after Swap are First number {firstNumber} and Second number {secondNumber}");  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
| 1. C# Program: SWAP NUMBERS WITHOUT using THIRD VARIABLE |
| Code: |
| using System;  namespace SwapNumber  {  internal class Program  {  static void Main(string[] args)  {  int firstNumber, secondNumber;  Console.Write("Enter First number: ");  firstNumber = Convert.ToInt32(Console.ReadLine());  Console.Write("Enter Second number: ");  secondNumber = Convert.ToInt32(Console.ReadLine());  Console.WriteLine($"Numbers before Swap are First number {firstNumber} and Second number {secondNumber}");  firstNumber = firstNumber + secondNumber;  secondNumber = firstNumber - secondNumber;  firstNumber = firstNumber - secondNumber;  Console.WriteLine($"Numbers after Swap are First number {firstNumber} and Second number {secondNumber}");  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
| 1. C# Program: Print Stars (\*) in Pattern - 1 [Right angled triangle pattern] |
| Code: |
| using System;  namespace Pattern  {  internal class Program  {  static void Main(string[] args)  {  int num;  Console.Write("Enter Number: ");  num = Convert.ToInt32(Console.ReadLine());  for (int i = 1; i <= num; i++)  {  for (int j = 1; j <= i; j++)  {  Console.Write("\*");  }  Console.WriteLine();  }  Console.ReadLine();  }  }  } |
| Output: |
|  |