|  |
| --- |
| **20 C# Programs Assignments**  **By**  **Bhanu Rama Krishna Prakash Jakkamsetti**  **27/1/20222** |

|  |
| --- |
| Program 1 |
| Print MULTIPLICATION TABLE of given number: |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace multiplicatio\_by\_using\_for  {  internal class Program2z  {  static void Main(string[] args)  {  int input, i;  Console.WriteLine("enter value");  input = Convert.ToInt32(Console.ReadLine());  for (i = 1; i <= 10; i++)  {  Console.WriteLine($"{input}\*{i}={input \* i}");  }  }  }  } |
| Output |
|  |
|  |
|  |

|  |
| --- |
| Program 2 |
| Print factorial of given number |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace factorial\_of\_numbers\_by\_using\_for  {  internal class Program  {  static void Main(string[] args)  {  int input, product = 1;  Console.WriteLine("Enter number");  input = Convert.ToInt32(Console.ReadLine());  for (int i = 1; i <= input; i++)  {  product = product \* i;  }  Console.WriteLine(product);  Console.ReadLine();  }  }  } |
| Output |
|  |

|  |
| --- |
| Program 3 |
| Print sum of N natural numbers |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace print\_sum\_of\_n\_natural\_numbers  {  internal class Program  {  static void Main(string[] args)  {  int input, sum = 0, i;  Console.WriteLine("Enter any value");  input = Convert.ToInt32(Console.ReadLine());  for (i = 1; i <= input; i++)  {  sum = sum + i;  }  Console.WriteLine(sum);  Console.ReadLine();  }  }  } |
| Output |
|  |

|  |
| --- |
| Program 4 |
| Print factorial using funtion |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace print\_factorial\_using\_function  {  internal class Program  {  static void Main(string[] args)  {  int input, i, fact = 1;  Console.WriteLine("Enter any number:");  input = Convert.ToInt32(Console.ReadLine());  getFact(input);  }  static void getFact(int input)  {  int fact = 1;  for (int i = 1; i <= input; i++)  {  fact = fact \* i;  }  Console.WriteLine("factorial of a given number:" + fact);  Console.ReadLine();  }  }  } |
| Output |
|  |

|  |
| --- |
| Program 5 |
| Print factorial using recursion |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace factorial\_using\_recursion  {  internal class Program  {  static void Main(string[] args)  {  Console.Write("Enter a Number : ");  int number = int.Parse(Console.ReadLine());  long factorial = RecursiveFactorial(number);  Console.Write($"Factorial of {number} is: {factorial}");  Console.ReadLine();  }  static long RecursiveFactorial(int number)  {  if (number == 1)  {  return 1;  }  else  {  return number \* RecursiveFactorial(number - 1);  }  }  }  } |
| Output |
|  |

|  |
| --- |
| Program 6 |
| Print factors using given number |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace print\_factors\_using\_given\_numbers  {  internal class Program  {  static void Main(string[] args)  {  int input, i;  Console.WriteLine("enter value");  input = Convert.ToInt32(Console.ReadLine());  for (i = 1; i <= input; i++)  {  if (input % i == 0)  {  Console.WriteLine(i);  }  }  }  }  } |
| Output |
|  |

|  |
| --- |
| Program 7 |
| Print power of given number [a power b] |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace print\_power\_of\_given\_number\_a\_power\_b  {  internal class Program  {  static void Main(string[] args)  {  int fn, sn, sum = 0;  int p = 1;  fn = 60;  Console.WriteLine("Enter First Number");  fn = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter Second Number");  sn = Convert.ToInt32(Console.ReadLine());  for (int i = 1; i <= sn; i++)  p = p \* fn;  Console.WriteLine("Power = " + p);  Console.ReadLine();  }  }  } |
| Output |
|  |

|  |
| --- |
| Program 8 |
| Prime numbers or not |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace prime\_number\_or\_not  {  internal class Program  {  static void Main(string[] args)  {  int input, count = 0, i;  Console.WriteLine("Enter any value");  input = Convert.ToInt32(Console.ReadLine());  for ( i = 1; i <=input; i++)  {  if (input%i == 0)  count++;  }  if (count==2)  {  Console.WriteLine($"{input} is a prime number");  }  else  {  Console.WriteLine($"{input} is not a prime number");  }  Console.ReadLine();  }  }  } |
| Output |
|  |
| Program 9 |
| Prime numbers check[using funtion] |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace prime\_check\_using\_function  {  internal class Program  {  static void Main(string[] args)  {  Console.WriteLine("Enter any number:");  int input = Convert.ToInt32(Console.ReadLine());  if (isPrimeNumber(input))  Console.WriteLine("It is a PrimeNumber", input);  else  Console.WriteLine("It is not a PrimeNumber", input);  Console.ReadLine();  }  static bool isPrimeNumber(int input)  {  for (int i = 2; i < input; i++)  {  if (input % i == 0)  {  return false;  }  }  return true;  }  }  } |
| Output |
|  |

|  |
| --- |
| Program 10 |
| Prime numbers in range |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace prime\_numbers\_in\_range  {  internal class Program  {  static void Main(string[] args)  {  Console.WriteLine("Enter number 1:");  int input1 = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter number 2:");  int input2 = Convert.ToInt32(Console.ReadLine());  for (int i = input1; i <= input2; i++)  {  isPrime(i);  }  Console.ReadLine();  }  static void isPrime(int input)  {  bool isPrimenum = true;  for (int i = 2; i < input; i++)  {  if (input % i == 0)  {  isPrimenum = false;  }  }  if (isPrimenum == true)  {  Console.WriteLine(input);  }  }  }  } |
| Output |
|  |

|  |
| --- |
| Program 11 |
| Fibonacci series |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace febunacci\_series  {  internal class Program  {  static void Main(string[] args)  {  int n, i, a = 0, b = 1, c;  Console.WriteLine("enter any number");  n = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("fibanocci series:0 1");  for ( i = 1; i <= n; i++)  {  c = a + b;  Console.WriteLine(c);  a = b;  b = c;    }  Console.ReadLine( );  }  }  } |
| Output |
|  |

|  |
| --- |
| Program 12 |
| Armstrong number |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace amstrong  {  internal class Program  {  static void Main(string[] args)  {  int n, r, sum = 0, temp;  Console.WriteLine("Enter the Number= ");  n = int.Parse(Console.ReadLine());  temp = n;  while (n > 0)  {  r = n % 10;  sum = sum + (r \* r \* r);  n = n / 10;  }  if (temp == sum)  Console.WriteLine("Armstrong Number.");  else  Console.WriteLine("Not Armstrong Number.");  }  }  } |
| Output |
|  |

|  |
| --- |
| Program 13 |
| Armstrong number [using function] |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace amstrong\_numbers\_using\_function  {  internal class Program  {  static void Main(string[] args)  {  int n, rem, m, res = 0;  Console.WriteLine("Enter any number :");  n = Convert.ToInt32(Console.ReadLine());  getArmtrong(n);  Console.ReadLine();  }  static void getArmtrong(int n)  {  int rem, m, res = 0;  m = n;  while (m > 0)  {  rem = m % 10;  m /= 10;  res = res + rem \* rem \* rem;  }  Console.WriteLine((res == n) ? "Armstrong" : "not Armstrong");  }  }  } |
| Output |
|  |

|  |
| --- |
| Program 14 |
| Armstrong number in range |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace amstrong\_range  {  internal class Program  {  static void Main(string[] args)  {  Console.WriteLine("Enter number 1:");  int input1 = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter number 2:");  int input2 = Convert.ToInt32(Console.ReadLine());  for (int i = input1; i <= input2; i++)  {  if (getArmtrong(i))  {  Console.WriteLine(i);  }  }  Console.ReadLine();  }  static bool getArmtrong(int n)  {  int rem, m, res = 0;  m = n;  while (m > 0)  {  rem = m % 10;  m /= 10;  res = res + rem \* rem \* rem;  }  return ((res == n) ? true : false);  }  }  } |
| Output |
|  |

|  |
| --- |
| Program 15 |
| Sum of digits of given number |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace sum\_of\_digits  {  internal class Program  {  static void Main(string[] args)  {  int n, sum = 0, m;  Console.WriteLine("Enter a number: ");  n = int.Parse(Console.ReadLine());  while (n > 0)  {  m = n % 10;  sum = sum + m;  n = n / 10;  }  Console.WriteLine($"Sum is= {sum}");  }  }    } |
| Output |
|  |

|  |
| --- |
| Program 16 |
| Reverse of given number |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace reverse\_of\_number  {  internal class Program  {  static void Main(string[] args)  {  int n, reverse = 0, rem;  Console.WriteLine("Enter a number: ");  n = int.Parse(Console.ReadLine());  while (n != 0)  {  rem = n % 10;  reverse = reverse \* 10 + rem;  n /= 10;  }  Console.WriteLine("Reversed Number: " + reverse);  }  }  } |
| Output |
|  |

|  |
| --- |
| Program 17 |
| Palindrome number |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace palindrome  {  internal class Program  {  static void Main(string[] args)  {  int n, r, sum = 0, temp;  Console.WriteLine("Enter the Number: ");  n = int.Parse(Console.ReadLine());  temp = n;  while (n > 0)  {  r = n % 10;  sum = (sum \* 10) + r;  n = n / 10;  }  if (temp == sum)  Console.WriteLine("Number is Palindrome.");  else  Console.WriteLine("Number is not Palindrome");  }  }  } |
| Output |
|  |

|  |
| --- |
| Program 18 |
| Swap numbers using third variable |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace swap\_numbers\_\_using\_third\_variable  {  internal class Program  {  static void Main(string[] args)  {  int a = 5, b = 10, t;  Console.WriteLine($"Before swap a={a} b={b}");  t = a;  a = b;  b = t;  Console.WriteLine($"After swap a={a} b={b}");  }  }  } |
| Output |
|  |

|  |
| --- |
| Program 19 |
| SWAP numbers without using third variable |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace swap\_numbers\_without\_using\_third\_variable  {  internal class Program  {  static void Main(string[] args)  {  int a = 5, b = 10;  Console.WriteLine($"Before swap a={a} b={b}");  a = a + b; //a=15 (5+10)  b = a - b; //b=5 (15-10)  a = a - b; //a=10 (15-5)  Console.WriteLine($"After swap a={a} b={b}");  }  }  } |
| Output |
|  |

|  |
| --- |
| Program 20 |
| Print stars(\*) in pattern1[right angled triangle] |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace print\_stars\_\_\_in\_pattern\_right\_angled\_triangle\_  {  internal class Program  {  static void Main(string[] args)  {  int n,i,j;  Console.WriteLine("enter no.of rows to be printed");  n= Convert.ToInt32(Console.ReadLine());  for (i = 1; i <=n; i++)  {  for (j = 1; j <=i; j++)  {  Console.Write("\* ");  }  Console.Write(" ");  Console.WriteLine();  }    }  }  } |
| Output |
|  |