Chapter 6

1. Write a program that prints on the console **the numbers from 1 to N**. The number **N** should be read from the standard input.

using System;

namespace \_01\_writingNloop

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Shkruaj Nje numer ");

int num = Convert.ToInt32(Console.ReadLine());

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

{

Console.WriteLine(i);

}

}

}

}

1. 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;

namespace \_02\_devisebleBy7

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Shkruaj numrin ");

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

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

{

if (i % (3 \* 7) != 0) Console.WriteLine(i);

}

}

}

}

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

using System;

namespace \_03\_pritingStoL

{

class Program

{

static void Main(string[] args)

{

int lowest = 0, highest = 0, input;

Console.Write("Enter numbers length: ");

int lenght = Int32.Parse(Console.ReadLine());

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

{

Console.Write("Enter number: ");

input = Int32.Parse(Console.ReadLine());

if (i == 0) lowest = highest = input;

else

{

if (lowest > input) lowest = input;

if (highest < input) highest = input;

}

}

Console.WriteLine("Lowest - {0}, Highest - {1}", lowest, highest);

}

}

}

1. 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;

namespace \_04\_cardDeck

{

class Program

{

static void Main(string[] args)

{

for (int i = 0; i < 4; i++)

{

if (i != 0) Console.WriteLine();

for (int j = 0; j < 13; j++)

{

switch (i)

{

case 0: Console.Write("Hearts "); break;

case 1: Console.Write("Diamonds "); break;

case 2: Console.Write("Spades "); break;

case 3: Console.Write("Clubs "); break;

}

switch (j)

{

case 0: Console.WriteLine("2"); break;

case 1: Console.WriteLine("3"); break;

case 2: Console.WriteLine("4"); break;

case 3: Console.WriteLine("5"); break;

case 4: Console.WriteLine("6"); break;

case 5: Console.WriteLine("7"); break;

case 6: Console.WriteLine("8"); break;

case 7: Console.WriteLine("9"); break;

case 8: Console.WriteLine("10"); break;

case 9: Console.WriteLine("J"); break;

case 10: Console.WriteLine("Q"); break;

case 11: Console.WriteLine("K"); break;

case 12: Console.WriteLine("A"); break;

}

}

}

}

}

}

1. Write a program that reads from the console number N and print the sum of the first N members of the **Fibonacci sequence**: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, …

using System;

namespace \_05\_fibonaciN

{

class Program

{

static void Main(string[] args)

{

int firstN = 0, secondN = 1, thirdN = 0;

Console.Write("Enter N: ");

int length = Int32.Parse(Console.ReadLine());

Console.Write("0, 1,");

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

{

thirdN = firstN + secondN;

Console.Write(" {0},", thirdN);

firstN = secondN;

secondN = thirdN;

}

}

}

}

1. Write a program that calculates **N!/K!** for given N and K (1<K<N).

using System;

namespace \_06\_factorialNk

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter N: (1<K<N) ");

int n = Int32.Parse(Console.ReadLine());

Console.Write("Enter K: (1<K<N) ");

int k = Int32.Parse(Console.ReadLine());

for (int i = n - 1; i > 0; i--)

{

n \*= i;

}

for (int i = k - 1; i > 0; i--)

{

k \*= i;

}

n /= k;

Console.WriteLine("Result is {0}", n);

}

}

}

1. Write a program that calculates **N!\*K!/(N-K)!** for given N and K (1<K<N).

using System;

namespace \_07\_NKNK

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter N: (1<K<N) ");

int n = Int32.Parse(Console.ReadLine());

Console.Write("Enter K: (1<K<N) ");

int k = Int32.Parse(Console.ReadLine());

int nMinusK = n - k;

for (int i = n - 1; i > 0; i--) n \*= i;

for (int i = k - 1; i > 0; i--) k \*= i;

for (int i = nMinusK - 1; i > 0; i--) nMinusK \*= i;

Console.WriteLine("Result is {0}", n \* k / nMinusK);

}

}

}

1. In combinatorics, the **Catalan numbers** are calculated by the following formula: [clip_image013[6]](https://introprogramming.info/wp-content/uploads/2013/07/clip_image0136.png), for n ≥ 0. Write a program that calculates the nth Catalan number by given n.

using System;

namespace \_08\_comBinatorics

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter N: (N >=0 ) ");

int n = Int32.Parse(Console.ReadLine());

int fact2N = 2 \* n, factNplus1 = n + 1;

for (int i = fact2N - 1; i > 0; i--) fact2N \*= i;

for (int i = factNplus1 - 1; i > 0; i--) factNplus1 \*= i;

for (int i = n - 1; i > 0; i--) n \*= i;

Console.WriteLine("Result is {0}", fact2N / (factNplus1 \* n));

}

}

}

1. Write a program that for a given integers **n**and**x**, **calculates the sum**: [clip_image015[6]](https://introprogramming.info/wp-content/uploads/2013/07/clip_image0156.png)

using System;

namespace \_09\_integerXK

{

class Program

{

static void Main(string[] args)

{

int sum = 1, temp = 1;

Console.Write("Enter n: ");

int n = Int32.Parse(Console.ReadLine());

Console.Write("Enter x: ");

int x = Int32.Parse(Console.ReadLine());

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

{

temp \*= i / x;

sum += temp;

}

Console.WriteLine("Result is {0}", sum);

}

}

}

1. Write a program that reads from the console a **positive integer number N** (N < 20) and prints a **matrix** of numbers as on the figures below:

using System;

namespace \_10\_positiveIntg

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter N: (N < 20) ");

int n = Int32.Parse(Console.ReadLine());

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

{

for (int j = i; j <= i; j++)

{

Console.Write("{0} ", j);

}

Console.WriteLine();

}

}

}

1. Write a program that calculates with **how many zeroes the factorial of a given number ends**. Examples:

N = 10 -> N! = 36288**00** -> 2

      N = 20 -> N! = 243290200817664**0000** -> 4

using System;

namespace \_11\_calculateZeros

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter N: ");

decimal n = Int32.Parse(Console.ReadLine());

int zeroes = 0;

for (int i = (int)(n - 1); i > 0; i--)

n \*= i;

Console.Write("N! is {0} and it ends ", n);

do

{

n /= 10;

zeroes++;

} while (n % 10 == 0);

Console.WriteLine("with {0} zeroes.", zeroes);

}

}

}

1. Write a program that converts a given number **from decimal to binary notation** (numeral system).

using System;

namespace \_12\_fromDeciBin

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter number: ");

int n = Int32.Parse(Console.ReadLine());

string binary = Convert.ToString(n, 2);

Console.WriteLine("Result is {0}", binary) }

}

}

1. Write a program that converts a given number **from binary to decimal notation**

using System;

namespace \_13\_fromBinDec

{

class Program

{

static void Main(string[] args)

{

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

int n = Int32.Parse(Console.ReadLine());

string toDecimal = Convert.ToString(Convert.ToInt32(n, 2), 10);

Console.WriteLine("Result is {0}", toDecimal);

}

}

}

1. Write a program that converts a given number **from** **decimal to hexadecimal notation**.

using System;

namespace \_14\_fromDeciHex

{

class Program

{

static void Main(string[] args)

{

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

int n = Int32.Parse(Console.ReadLine());

string toDecimal = Convert.ToString(Convert.ToInt32(n, 10), 16);

Console.WriteLine("Result is {0}", toDecimal);

}

}

}

1. Write a program that converts a given number **from hexadecimal to decimal notation**.

using System;

namespace \_15\_fromHexDeci

{

class Program

{

static void Main(string[] args)

{

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

int n = Int32.Parse(Console.ReadLine());

string toDecimal = Convert.ToString(Convert.ToInt32(n, 16), 10);

Console.WriteLine("Result is {0}", toDecimal);

}

}

}

1. Write a program that by a given integer **N** prints the numbers from 1 to N in **random order**.

using System;

namespace \_16\_1Random

{

class Program

{

static void Main(string[] args)

{

Random rnd = new Random();

int temp, randomNumber;

Console.Write("Enter number: ");

int n = Int32.Parse(Console.ReadLine());

int[] arr = new int[n];

for (int i = 0; i < arr.Length; i++)

{

arr[i] = i;

}

foreach (int i in arr)

{

randomNumber = rnd.Next(0, n);

temp = arr[i];

arr[i] = arr[randomNumber];

arr[randomNumber] = temp;

}

foreach (int i in arr) Console.WriteLine(arr[i]);

}

}

}

1. Write a program that given two numbers finds their **greatest common divisor (GCD)** and their **least common multiple (LCM)**. You may use the formula **LCM(a, b) = |a\*b| / GCD(a, b)**.

using System;

namespace \_17\_greatesDeviser

{

class Program

{

static void Main(string[] args)

{

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

int a = Int32.Parse(Console.ReadLine());

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

int b = Int32.Parse(Console.ReadLine());

while (a != 0 && b != 0)

{

if (a > b) a %= b;

else b %= a;

}

if (a == 0) Console.WriteLine(b);

else Console.WriteLine(a);

}

}

}

1. \* Write a program that for a given number n, outputs a matrix in the form of a **spiral**:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 12 | 13 | 14 | 5 |
| 11 | 16 | 15 | 6 |
| 10 | 9 | 8 | 7 |

Example for n=4:

using System;

namespace \_18\_nMatrix

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter N: ");

int n = Int32.Parse(Console.ReadLine());

int[,] matrix = new int[n, n];

int row = 0, col = 0, direction = 0;

for (int i = 1; i <= n \* n; i++)

{

switch (direction)

{

case 0:

if (col > n - 1 || matrix[row, col] != 0)

{

direction = 1;

col--;

row++;

}

break;

case 1:

if (row > n - 1 || matrix[row, col] != 0)

{

direction = 2;

row--;

col--;

}

break;

case 2:

if (col < 0 || matrix[row, col] != 0)

{

direction = 3;

col++;

row--;

}

break;

case 3:

if (row < 0 || matrix[row, col] != 0)

{

direction = 0;

row++;

col++;

}

break;

}

matrix[row, col] = i;

switch (direction)

{

case 0: col++; break;

case 1: row++; break;

case 2: col--; break;

case 3: row--; break;

}

}

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

{

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

{

if (matrix[i, j] < 10) Console.Write("{0} ", matrix[i, j]);

else Console.Write("{0} ", matrix[i, j]);

}

Console.WriteLine();

}

}

}

}