

Detyra te shtepise

Hyrje ne Struktura e te te Dhenave

Studenti/ja: Gerti Gonxhi

Kampusi Prishtine

Viti I-Grupi II

Ligjeruesi: Laberion Zebica

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 ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

int n;

Console.Write("Enter an integer number:");

bool isnInt = int.TryParse(Console.ReadLine(), out n);

if (isnInt)

{

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

{

Console.WriteLine(i);

}

}

else

{

Console.WriteLine("Not a valid entry!");

}

}

}

}

2.Write a program that prints on the console the numbers form 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 ConsoleApp2

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter N: ");

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

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

{

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

}

}

}

}

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

using System;

namespace ConsoleApp3

{

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);

}

}

}

4.Write a program that prints all possible cards from a standard deck of cards,without jokers(there are 53 cards: 4 suits of 13 cards).

using System;

namespace ConsoleApp4

{

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;

}

}

}

Console.ReadLine();

}

}

}

5.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 ConsoleApp5

{

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;

}

}

}

}

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

using System;

namespace ConsoleApp6

{

class Program

{

static void Main(string[] args)

{

int N, K;

int divisor = 1;

double result;

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

bool isNInt = int.TryParse(Console.ReadLine(), out N);

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

bool isKInt = int.TryParse(Console.ReadLine(), out K);

if (isNInt && isKInt && (N < K) && (N > 1) && (K > 1))

{

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

{

divisor = divisor \* i;

}

result = 1.0 / (double)divisor;

Console.WriteLine("result={0}", result);

}

else

{

Console.WriteLine("Not a valid entry!");

}

}

}

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

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Numerics;

namespace ConsoleApp7

{

class Program

{

static void Main(string[] args)

{

int N, K;

BigInteger result = 1;

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

bool isNInt = int.TryParse(Console.ReadLine(), out N);

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

bool isKInt = int.TryParse(Console.ReadLine(), out K);

if (isNInt && isKInt && (N < K) && (N > 1) && (K > 1))

{

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

{

if (N > (K - N))

{

result = result \* i;

if (i < (N + 1))

{

result = result \* i;

}

}

else if (N < (K - N))

{

if (i <= N)

{

result = result \* i;

}

if (i > (K - N))

{

result = result \* i;

}

}

else//N=K-N

{

result = result \* i;

}

}

Console.WriteLine("result={0}", result);

}

else

{

Console.WriteLine("Not a valid entry!");

}

}

} }

8.   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 ConsoleApp8

{

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));

}

}

}

9.    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;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Numerics;

namespace ConsoleApp8

{

class Program

{

private static BigInteger FindFactorial(int N)

{

BigInteger factorial = 1;

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

{

factorial = factorial \* i;

}

return factorial;

}

private static BigInteger FindPower(int X, int N)

{

BigInteger powerX = 1;

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

{

powerX = powerX \* X;

}

return powerX;

}

static void Main()

{

int N, X;

double sum = 1.0;

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

bool isNInt = int.TryParse(Console.ReadLine(), out N);

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

bool isXInt = int.TryParse(Console.ReadLine(), out X);

if (isNInt && isXInt)

{

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

{

sum = sum + ((double)FindFactorial(i) / (double)FindPower(X, i));

}

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

}

else

{

Console.WriteLine("Not a valid entry!");

}

}

}

}

10.   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:

**N = 3**                         **N = 4**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | 1 | 2 | 3 | | 2 | 3 | 4 | | 3 | 4 | 5 | | |  |  |  |  | | --- | --- | --- | --- | | 1 | 2 | 3 | 4 | | 2 | 3 | 4 | 5 | | 3 | 4 | 5 | 6 | | 4 | 5 | 6 | 7 | |

using System;

namespace ConsoleApp10

{

class Program

{

static void Main(string[] args)

{

int N;

Console.Write("Enter an integer number < 20:");

bool isnInt = int.TryParse(Console.ReadLine(), out N);

if (isnInt & N < 20 & N > 0)

{

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

{

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

{

if (j + i > 9)

{

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

}

else

{

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

}

}

Console.WriteLine();

}

}

else

{

Console.WriteLine("Wrong entry!");

}

}

}

}

11.   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;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Numerics;

namespace ConsoleApp11

{

class Program

{

static void Main(string[] args)

{

int N, temporary;

int numberOfZeroes = 0;

BigInteger factorial = 1;

Console.Write("Enter an integer number:");

bool isnInt = int.TryParse(Console.ReadLine(), out N);

if (isnInt)

{

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

{

factorial = factorial \* i;

}

Console.WriteLine("N factorial={0}", factorial);

if (N >= 5)

{

do

{

temporary = N / 5;

if (temporary != 0)

{

numberOfZeroes = numberOfZeroes + temporary;

N = temporary;

}

else

{

break;

}

}

while (true);

}

Console.WriteLine("Number of zeroes =:{0}", numberOfZeroes);

}

else

{

Console.WriteLine("Wrong entry!");

}

} } }

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

using System;

namespace ConsoleApp12

{

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);

}

}

}

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

using System;

namespace ConsoleApp13

{

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), 10);

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

}

}

}

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

using System;

namespace ConsoleApp14

{

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);

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

}

}

}

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

using System;

namespace ConsoleApp15

{

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);

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

}

}

}

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

using System;

namespace ConsoleApp16

{

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]);

}

}

}

17.   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 ConsoleApp17

{

class Program

{

static void Main(string[] args)

{

uint N, K, temporary, remaining;

uint greatestCommonDevider = 1;

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

bool isNUint = uint.TryParse(Console.ReadLine(), out N);

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

bool isKUint = uint.TryParse(Console.ReadLine(), out K);

if (isNUint && isKUint && (N != 0 || K != 0))

{

if (N == 0)

{

greatestCommonDevider = K;

}

else if (K == 0)

{

greatestCommonDevider = N;

}

else

{

temporary = Math.Max(N, K);

K = Math.Min(N, K);

N = temporary;

do

{

remaining = N % K;

if (remaining == 0)

{

greatestCommonDevider = K;

break;

}

else

{

N = K;

K = remaining;

}

} while (true);

}

Console.WriteLine("Greatest common devider = {0}", greatestCommonDevider);

}

else

{

Console.WriteLine("Not a valid entry!");

}

}

}

}

18.   \* 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 ConsoleApp18

{

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();

}

}

}

}