**Aplicatii suplimentare**

**Functii**

**Problema 1**

using System;

namespace P1

{

class Program

{

/// <summary>

/// Scrieţi definiţia completă a subprogramului sub cu doi parametri:

/// n (număr natural, 0<n≤50) şi k (număr natural, 0<k≤20).

/// Subprogramul determină afişarea pe o linie nouă a ecranului,

/// în ordine descrescătoare, a primelor n numere naturale nenule divizibile cu k.

/// Numerele vor fi separate prin câte spaţiu.

/// Exemplu: dacă n=3 şi k=5 la apelul subprogramului se va afişa pe ecran: 15 10 5.

/// </summary>

static void Main(string[] args)

{

int n, k;

Console.WriteLine("Introduceti n: ");

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

Console.WriteLine("Introduceti k: ");

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

sub(n, k);

}

private static void sub(int n, int k)

{

//primele n numere naturale nenule divizibile cu k <=> primii n multipli ai lui k

//acesti multipli pot fi obtinuti prin inmultirea lui k cu 1, 2, ..., n

//pentru a-i putea afisa descrescator, incep inmultirea lui k cu n, n-1, ..., 1

Console.WriteLine("Primele n numere divizibile cu k, afisate descrescator, sunt: ");

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

Console.Write(k\*i+" ");

}

}

}

**Problema 2**

using System;

namespace P2

{

class Program

{

/// <summary>

/// Scrieţi definiţia completă a subprogramului sub cu trei parametri:

/// n (număr natural, 5<n≤30000), a şi b; subprogramul furnizează prin intermediul

/// parametrilor a şi b cele mai mari două numere prime distincte mai mici decât n.

/// Exemplu: dacă n= 28 la apelul subprogramului se va furniza prin parametrul a valoarea 23 şi prin parametrul b valoarea 19.

/// </summary>

static void Main(string[] args)

{

int n,a=-1,b=-1;

Console.WriteLine("Introduceti n: ");

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

sub(n, ref a, ref b);

Console.WriteLine("Primele doua numere prime mai mici decat n sunt: " + a +" "+ b);

}

private static void sub(int n, ref int a, ref int b)

{

bool prim=false; //prin intrmediul acestei variabile vom retine daca un nr este sau nu prim

n--; //deoarece numerele trebuie sa fie mai mici decat n

if (n % 2 == 0)

n--; //deoarece un numar par sigur nu e prim

for (int i=n;i>0 && prim==false;i-=2) //i merge din 2 in 2, descrescator, pt a sari peste numerele pare

{

prim = true; //presupunem ca i este nr prim, iar daca se va dovedi ca nu este, prim va redeveni false

for (int j = 3; j \* j <= i; j += 2)

if (i % j == 0)

{

prim = false; //daca i se imparte exact la j, inseamna ca nu este prim break; //putem merge mai departe

}

if (prim==true) //daca prim a ramas true inseamna ca nu s-a gasit niciun divizor pt i si se iese din for deoarece prim != false

a = i;

}

prim = false;

for(int i=a-2;i>0 && prim==false; i-=2) //cautam al doilea nr prim incepand cu a-2; comentariile de mai sus raman valabile si aici

{

prim = true;

for (int j = 3; j \* j <= i; j += 2)

if (i % j == 0)

{

prim = false;

break;

}

if (prim == true)

b = i;

}

}

}

}

**Problema** **3**

using System;

namespace P3

{

class Program

{

/// <summary>

/// Scrieţi definiţia completă a subprogramului multiplu care are 3 parametri:

/// a, prin care primeşte un tablou unidimensional cu maximum 100 de numere naturale mai mici decât 1000,

/// n, numărul efectiv de elemente ale tabloului şi k, un număr natural (k≤9).

/// Subprogramul returnează numărul de elemente din tablou care sunt multipli ai numărului k şi au ultima cifră egală cu k.

/// Exemplu: dacă n=6, a=(9,273,63,83,93,123), iar k=3, subprogramul va returna valoarea 4.

/// </summary>

static void Main(string[] args)

{

int n, k;

Console.WriteLine("Introduceti n (nr. de elemente): ");

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

int[] a = new int[n];

Console.WriteLine("Introduceti pe o singura linie elementele tabloului: ");

string[] buffer = Console.ReadLine().Split(' ');

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

a[i] = int.Parse(buffer[i]);

Console.WriteLine("Introduceti k: ");

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

Console.WriteLine("Nr de elemente care respecta cerinta este: "+multiplu(a,n,k));

}

private static int multiplu(int[] a, int n, int k)

{

int nrElemente = 0; //stocheaza nr de elemente care respecta cerinta

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

if (a[i] % k == 0 && a[i] % 10 == k) //verifica daca elementul de pe pozitia i este divizibil cu k si are ultima cifra k

nrElemente++;

return nrElemente;

}

}

}

**Problema 4**

using System;

namespace P4

{

class Program

{

/// <summary>

/// Scrieţi definiţia completă a subprogramului interval care are doi parametri a şi n,

/// prin care primeşte un tablou unidimensional cu maximum 100 de numere naturale mai mici decât 1000

/// şi respectiv numărul efectiv de elemente din tabloul unidimensional.

/// Subprogramul returnează numărul de elemente din tabloul unidimensional

/// care aparţin intervalului închis determinat de primul şi respectiv ultimul element al tabloului.

/// Exemplu: dacă tabloul are 6 elemente şi este de forma (12,27,6,8,9,2), subprogramul va returna valoarea 5.

/// </summary>

static void Main(string[] args)

{

int n;

Console.WriteLine("Introduceti n: ");

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

int[] a = new int[n];

Console.WriteLine("Introduceti elementele tabloului pe o singura linie: ");

string[] buffer = Console.ReadLine().Split(' ');

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

a[i] = int.Parse(buffer[i]);

Console.WriteLine("Nr de elemente care apartin intervalului este: " + interval(a,n));

}

private static int interval(int[] a, int n)

{

int nr1, nr2; //variabile in care se vor memora capetele tabloului

int nrElemente=0; //variabila in care se va stoca nr de elemente care verifica conditia din cerinta

//verific care dintre cele doua numere din capete este mai mare si stochez in nr1 numarul mai mic, iar in nr2 numarul mai mare

if(a[0]>a[n-1])

{

nr1 = a[n - 1];

nr2 = a[0];

}

else

{

nr1 = a[0];

nr2 = a[n - 1];

}

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

if (a[i] >= nr1 && a[i] <= nr2) //verific apartenenta la interval

nrElemente++;

return nrElemente;

}

}

}

**Problema 5**

using System;

namespace P5

{

class Program

{

/// <summary>

/// Scrieţi definiţia completă a subprogramului count care are doi parametri, a şi n,

/// prin care primeşte un tablou unidimensional cu maximum 100 de numere reale

/// şi respectiv numărul efectiv de elemente din tablou. Subprogramul returnează

/// numărul de elemente din tabloul a care sunt mai mari sau cel puţin egale

/// cu media aritmetică a tuturor elementelor din tablou.

/// Exemplu: dacă tabloul are 6 elemente şi este de forma (12, 7.5, 6.5, 3, 8.5, 7.5),

/// subprogramul va returna valoarea 4 (deoarece media tuturor elementelor este 7.5 şi

/// numerele subliniate sunt cel puţin egale cu această medie).6.

/// </summary>

static void Main(string[] args)

{

int n;

Console.WriteLine("Introduceti n: ");

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

float[] a = new float[n];

Console.WriteLine("Introduceti elementele tabloului pe o singura linie: ");

string[] buffer = Console.ReadLine().Split(' ');

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

a[i] = float.Parse(buffer[i]);

Console.WriteLine("Nr de elemente este: "+ count(a,n));

}

private static int count(float[] a, int n)

{

float suma = 0, media; //in suma se va stoca suma elementelor pentru a putea face media lor aritmetica,n in medie se va stoca media

int nrElemente = 0; //in nrElemente se va stoca nr de elemente care verifica consitia

for (int i = 0; i < n; i++) //parcurgem vectorul pentru a face suma elementelor

suma += a[i];

media = suma / n;

for (int i = 0; i < n; i++) //parcurgem vectorul pentru a verifica conditia din cerinta

if (a[i] >= media)

nrElemente++;

return nrElemente;

}

}

}

**Problema 6**

using System;

namespace P6

{

class Program

{

/// <summary>

/// Subprogramul aranjare are doi parametri: a prin care primeşte un tablou unidimensional

/// cu maximum 100 de numere reale nenule şi n, numărul de elemente din tablou.

/// Subprogramul rearanjează elementele tabloului unidimensional astfel încât toate

/// valorile negative să se afle pe primele poziţii, iar valorile pozitive în continuarea celor negative.

/// Ordinea în cadrul secvenţei de elemente pozitive, respectiv în cadrul secvenţei de elemente negative, poate fi oricare.

/// Tabloul modificat va fi furnizat tot prin intermediul parametrului a.

/// Exemplu: dacă tabloul are 6 elemente şi este de forma (12, -7.5, 6.5, -3, -8, 7.5), după apel, acesta ar putea fi: (-7.5, -3, -8, 12, 6.5, 7.5). Scrieţi definiţia completă a subprogramului aranjare

/// </summary>

static void Main(string[] args)

{

int n;

Console.WriteLine("Introduceti n: ");

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

double[] a = new double[n];

Console.WriteLine("Introduceti elementele tabloului pe o singura linie: ");

string[] buffer = Console.ReadLine().Split(' ');

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

a[i] = double.Parse(buffer[i]);

aranjare(ref a, n);

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

Console.Write(a[i]+" ");

}

private static void aranjare(ref double[] a, int n)

{

double temp;

for (int i = 0; i < n - 1; i++) //verifica mai intai primul element cu cele care urmeaza dupa el si, daca e pozitiv, le interschimba, ducand spre capat numerele pozitive; procedeaza la fel cu al doilea element si restul etc.

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

if (a[i] > 0)

{

temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

}

}

**Problema 7**

using System;

namespace P7

{

class Program

{

/// <summary>

/// Subprogramul nule are doi parametri: a, prin care primeşte un tablou unidimensional

/// cu maximum 100 de numere întregi, cu cel mult 4 cifre fiecare şi n, numărul de elemente din tablou.

/// Subprogramul rearanjează elementele tabloului unidimensional astfel încât toate valorile nule

/// să se afle la sfârşitul tabloului. Ordinea în cadrul secvenţei de elemente nenule poate fi oricare.

/// Tabloul modificat este furnizat tot prin parametrul a. Exemplu: dacă n=6, a=(12,0,0,-3,-8,0),

/// după apel, acesta ar putea fi: a=(12,-3,-8,0,0,0). Scrieţi definiţia completă a subprogramului nule.

/// </summary>

/// <param name="args"></param>

static void Main(string[] args)

{

int n;

Console.WriteLine("Introduceti n: ");

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

double[] a = new double[n];

Console.WriteLine("Introduceti elementele tabloului pe o singura linie: ");

string[] buffer = Console.ReadLine().Split(' ');

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

a[i] = double.Parse(buffer[i]);

nule(ref a, n);

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

Console.Write(a[i] + " ");

}

private static void nule(ref double[] a, int n)

{

double temp;

for (int i = 0; i < n - 1; i++) //verifica mai intai primul element cu cele care urmeaza dupa el si, daca e 0, le interschimba, ducand spre capat numerele nule; procedeaza la fel cu al doilea element si restul etc.

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

if (a[i] == 0)

{

temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

}

}