

Neptun kód: **ABC123** Név: **Készítő Neve**
 Beadás verziószáma: 1.
 Később kiegészítéssel(!)
 2. (javított Uf...)
 3. (javított Algoritmus)

Feladat – Szélcsendes napok száma

A vitorlás versenyek rendezői megmérték N napon keresztül Balatonfüreden a szélsébséget. Vitorlás versenyt szélcsendben nem lehet rendezni, de nagyobb viharban sem célszerű. A versenyen naponta egy fordulót rendeznek, s legalább K fordulóra van szükség. Írj programot, amely kiszámolja, hogy hány napon volt szélcsend!

Specifikáció

Be: $\text{napSzam} \in \mathbb{N}$, $\text{vNapSzam} \in \mathbb{N}$, $\text{szelErossegek} \in \mathbb{Z}[1..\text{napSzam}]$

Ki: $\text{db} \in \mathbb{N}$

Ef: $\text{napSzam} \geq 1$ és $\text{napSzam} \leq 1000$ és $\text{vNapSzam} \geq 1$ és $\text{vNapSzam} \leq 10$ és $\forall i \in [1..\text{napSzam}]: (0 \leq \text{szelErossegek}[i] \leq 200)$

Uf: $\text{db} = \text{DARAB}(i=1..\text{napSzam}, \text{szelErossegek}[i] = 0)$ és

$\text{vNapSzam} = \text{vNapSzam}$

Sablon – Megszámolás

Be: $e \in \mathbb{Z}$, $u \in \mathbb{Z}$

Ki: $\text{db} \in \mathbb{N}$

Ef: -

Uf: $\text{db} = \text{DARAB}(i=e..u, T(i))$

Megszámolás sablon

Feladat

Adott az egész számok egy $[e..u]$ intervalluma és egy $T:[e..u] \rightarrow \text{Logikai feltétel}$. Határozzuk meg, hogy az $[e..u]$ intervallumon a T feltétel **hányszor** veszi fel az igaz értéket!

Specifikáció

Be: $e \in \mathbb{Z}$, $u \in \mathbb{Z}$

Ki: $\text{db} \in \mathbb{N}$

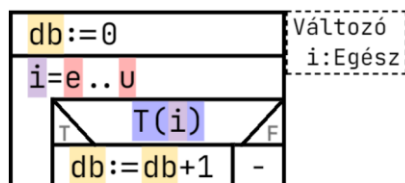
Ef: -

Uf: $\text{db} = \text{SZUMMA}(i=e..u, 1, T(i))$

Rövidítve:

Uf: $\text{db} = \text{DARAB}(i=e..u, T(i))$

Algoritmus



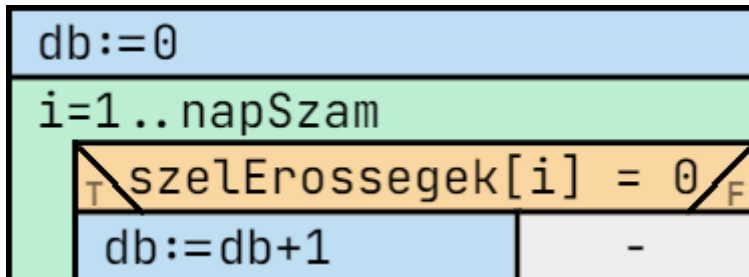
i	T(i)	érték
e	IGAZ	1
e+1	HAMIS	0
...	HAMIS	0
u	IGAZ	1
		db= 2

Visszavezetés

$i = e..u \quad \sim \quad i = 1..napSzam$

$T(i) \quad \sim \quad szelErossegek[i] = 0$

Algoritmus



Kód (C#)

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Project01_szelcsendesNapok_BPatrik_5cs
{
    internal class Program
    {
        static void Main(string[] args)
        {
            // deklarálás: spec be, ki

            int napSzam, vNapSzam, db;

            int[] szelErossegek;

            // beolvasás: spec be

            string sor = Console.ReadLine();

            string[] tomb = sor.Split(' ');

            napSzam = int.Parse(tomb[0]);

            vNapSzam = int.Parse(tomb[1]);
```

Megoldás sablon „C” beadandó 1. fázishoz (lecserélendő a konkrét beadandó megnevezésére)

```
szelErossegek = new int[napSzam];

for (int i = 0; i < napSzam; i++)
{
    szelErossegek[i] = int.Parse(Console.ReadLine());
}

// feldolgozás: alg, stuki

db = 0;

for (int i = 1; i <= napSzam; i++)
{
    if (szelErossegek[i-1] == 0)
    {
        db++;
    }
}

// kiírás: spec ki

Console.WriteLine(db);
}
}
}
```

Bíró pontszám és képernyőkép

TÉMAVÁLTÁS

BEAD

EREDMÉNY

LETÖLTÉS

VISSZATÖLT

BEADOTTAK

FŐOLDAL

KILÉP/BELÉP

Utolsó beadás eredménye

Összpont: 100/

Teszt#	Pont	Verdikt...	futási idő
1.1	3/3	Helyes	0.031 sec
2.1	3/3	Helyes	0.039 sec
3.1	3/3	Helyes	0.031 sec
4.1	3/3	Helyes	0.031 sec
5.1	3/3	Helyes	0.031 sec
6.1	3/3	Helyes	0.031 sec
7.1	3/3	Helyes	0.031 sec
8.1	3/3	Helyes	0.032 sec
9.1	4/4	Helyes	0.031 sec
10.1	4/4	Helyes	0.031 sec
11.1	4/4	Helyes	0.034 sec
12.1	4/4	Helyes	0.029 sec
13.1	4/4	Helyes	0.033 sec
14.1	4/4	Helyes	0.030 sec
15.1	4/4	Helyes	0.032 sec
16.1	4/4	Helyes	0.031 sec
17.1	4/4	Helyes	0.033 sec
18.1	4/4	Helyes	0.031 sec
19.1	4/4	Helyes	0.031 sec
20.1	4/4	Helyes	0.031 sec
21.1	4/4	Helyes	0.031 sec
22.1	4/4	Helyes	0.031 sec
23.1	4/4	Helyes	0.035 sec
24.1	4/4	Helyes	0.030 sec
25.1	4/4	Helyes	0.030 sec
26.1	4/4	Helyes	0.031 sec
27.1	4/4	Helyes	0.031 sec

Beadva: 2024-10-22 10:03:58.0

Saját tesztfájlok

be1.txt

Megoldás sablon „C” beadandó 1. fázishoz (lecserélendő a konkrét beadandó megnevezésére)

3 2

0

0

0

be2.txt

10 5

50

200

0

0

127

20

15

0

0

5

be3.txt

2 2

50 200

be4.txt

20 7

50

200

0

0

127

20

15

Megoldás sablon „C” beadandó 1. fázishoz (lecserélendő a konkrét beadandó megnevezésére)

0

0

5

0

0

0

1

40

120

0

90

100

0