

Neptun kód: **UW0FDO**  
Beadás verziószáma: 1

Név: **Szász Roland**

## Feladat

### A nap mikor nem volt kapás

Egy ember  $N$  napon horgászott, tudjuk, hogy melyik nap hány halat fogott.

Készíts programot, amely megad egy napot, amikor nem fogott halat!

#### Bemenet

A *standard bemenet* első sorában a napok száma van ( $1 \leq N \leq 100$ ). A következő sorok mindegyike egy-egy egész számot tartalmaz, az egyes napokon fogott halak számát ( $0 \leq H \leq 20$ ).

#### Kimenet

A *standard kimenet* első sorába egyetlen egész számot kell írni, a nap sorszámát, amikor nem fogott halat! Ha nincs ilyen nap, akkor a sorba -1-et kell kiírni! Több megoldás esetén a legkisebb sorszámút kell kiírni!

## Adatreprezentáció

1	2	3	4	5
<b># Bemenet</b> n: 6 kapasok: [20, 1, 0, 3, 10, 6] <b># Segédadat</b> van: true <b># Kimenet</b> ind: 3	<b># Bemenet</b> n: 4 kapasok: [5, 1, 10, 2] <b># Segédadat</b> van: false <b># Kimenet</b> ind: -1	<b># Bemenet</b> n: 5 kapasok: [0, 2, 5, 0, 1] <b># Segédadat</b> van: true <b># Kimenet</b> ind: 1	<b># Bemenet</b> n: 3 kapasok: [5, 10, 0] <b># Segédadat</b> van: true <b># Kimenet</b> ind: 3	<b># Bemenet</b> n: 3 kapasok: [5, 21, 2] <b># Segédadat</b> van: (nem értelmezett) <b># Kimenet</b> ind: -
Megadva	Nincs Találat	Helyes	Helyes	Előfeltétel hiba

## Specifikáció

Be:  $n \in \mathbb{N}$ ,  $\text{kapasok} \in \mathbb{N}[1..n]$

Sa:  $\text{van} \in \mathbb{L}$

Ki:  $\text{ind} \in \mathbb{Z}$

Ef:  $1 \leq n \leq 100$  és

$\forall i \in [1..n]: (0 \leq \text{kapasok}[i] \leq 20)$

Uf:  $(\text{van}, \text{ind}) = \text{KERES}(i=1..n, \text{kapasok}[i] = 0)$  és  
 $(\text{nem van}) \rightarrow (\text{ind} = -1)$

Link: [Specification editor](#)

# Keresés sablon

## Feladat

Adott az egész számok egy  $[e..u]$  intervalluma és egy  $T:[e..u] \rightarrow \text{Logikai feltétel}$ . Határozzuk meg az  $[e..u]$  intervallumban balról az első olyan számot, ha van, amely kielégíti a  $T$  feltételt!

## Specifikáció

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

Ki:  $\text{van} \in \mathbb{L}, \text{ind} \in \mathbb{Z}$

Ef: -

Uf:  $\text{van} = \exists i \in [e..u] : (T(i))$  és  
 $\text{van} \rightarrow (\text{ind} \in [e..u] \text{ és } T(\text{ind}) \text{ és } \forall i \in [e..\text{ind}-1] : (\text{nem } T(i)))$

Rövidítve:

Uf:  $(\text{van}, \text{ind}) = \text{KERES}(i=e..u, T(i))$

## Algoritmus

$i := e$	Változó $i: \text{Egész}$
$i \leq u$ és nem $T(i)$	
$i := i + 1$	
$\text{van} := i \leq u$	
$\text{ind} := i$	
$\text{van}$	F
-	T



## Visszavezetés

Megszámolás		
$i$	$\sim$	$i$
$e..u$	$\sim$	$1..n$
$T(i)$	$\sim$	$\text{kapasok}[i] = 0$
$(\text{van}, \text{ind})$	$\sim$	$(\text{van}, \text{ind})$

## Algoritmus

Local declarations

n, ind: Egész

kapasok: Tömb[1..n:Egész]

van: Logikai

i: Egész

Be: n, kapasok

van:=hamis

i:=1

nem van és  $i \leq \text{hossz}(\text{kapasok})$

van:=kapasok[i]=0

ind:=i

i:=i+1

T / nem van / F

ind:= -1

-

Ki: ind

Link: [Structogram editor](#)

## 2. fázishoz

### Kód

```
// Név: Szász Roland
// Neptun kód: UW0FDO
// Inf-es e-mail: UW0FDO@inf.elte.hu

using System;

class Program {
    static void Main(string[] args) {
        #region Deklaráció

        int n;
        int ind;
        int[] kapasok;
        bool van;
        int i;

        #endregion

        #region Beolvasás

        n = int.Parse(Console.ReadLine());
        kapasok = new int[n];

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

        #endregion

        #region Feldolgozás

        van = false;
        i = 0;
        ind = 0;

        while (!(van) && i < kapasok.Length) {
            van = kapasok[i] == 0;
            ind = i+1;
            ++i;
        }

        if (!(van)) {
            ind = -1;
        }

        #endregion

        #region Kiírás
        Console.WriteLine($"{ind}");
    }
}
```

```
}  
    #endregion  
}
```

## Bíró eredmény

Verzió: 1.0

Összpont: 100/100

Teszt#	Pont	...Verdikt...	futási idő
1.1	3/3	Helyes	0.028 sec
2.1	3/3	Helyes	0.032 sec
3.1	3/3	Helyes	0.029 sec
4.1	3/3	Helyes	0.030 sec
5.1	3/3	Helyes	0.030 sec
6.1	3/3	Helyes	0.030 sec
7.1	3/3	Helyes	0.031 sec
8.1	3/3	Helyes	0.029 sec
9.1	4/4	Helyes	0.030 sec
10.1	4/4	Helyes	0.030 sec
11.1	4/4	Helyes	0.029 sec
12.1	4/4	Helyes	0.029 sec
13.1	4/4	Helyes	0.029 sec
14.1	4/4	Helyes	0.031 sec
15.1	4/4	Helyes	0.029 sec
16.1	4/4	Helyes	0.030 sec
17.1	4/4	Helyes	0.030 sec
18.1	4/4	Helyes	0.029 sec
19.1	4/4	Helyes	0.028 sec
20.1	4/4	Helyes	0.030 sec
21.1	4/4	Helyes	0.029 sec
22.1	4/4	Helyes	0.030 sec
23.1	4/4	Helyes	0.030 sec
24.1	4/4	Helyes	0.030 sec
25.1	4/4	Helyes	0.029 sec
26.1	4/4	Helyes	0.029 sec
27.1	4/4	Helyes	0.030 sec

Beadva: 2025-10-30 16:08:11.0