

Neptun kód:	UW0FDO	Név:	Szász Roland
Beadás verziószáma:	1		

Feladat

Névsorban növekvő magasság

Adott a tanulók neve és magassága névsor szerint rendezve.

Írj programot, amely eldönti, hogy a névsor és a magasság szerinti sorrend azonos-e!

Bemenet

A *standard bemenet* első sorában található a tanulók száma ($1 \leq N \leq 100$). A következő N sorban egy-egy szóból álló név és egy centiméterben megadott magasságértéket tartalmazó szám van szóközzel elválasztva.

Kimenet

A *standard kimenet* első sorába ki kell írni a választ! Az „IGEN” szöveg szerepeljen amennyiben a magasság is monoton növekszik, különben a „NEM” szöveg!

Adatreprezentáció

Szituáció 1	Szituáció 2
<pre># Bemenet t: 6 diakok: [{nev: "Anna", magassag: 168}, {nev: "Bea", magassag: 170}, {nev: "Csaba", magassag: 172}, {nev: "Dominik", magassag: 172}, {nev: "Erik", magassag: 176}, {nev: "Ferenc", magassag: 180}] # Segédadat van: false # Kimenet valasz: "IGEN"</pre>	<pre># Bemenet t: 0 diakok: [] # Segédadat van: false # Kimenet valasz: "??"</pre>
Megadva (Helyes)	Hibás

Specifikáció

Bé: $t \in \mathbb{N}$, $\text{diakok} \in \text{Diak}[1..t]$, $\text{Diak} = (\text{nev}: S \times \text{magassag}: N)$

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

Ki: $\text{valasz} \in S$

Ef: $1 \leq t \leq 100$

Uf: $(\text{van},) = \text{KERES}(i=1..t-1, \text{diakok}[i].\text{magassag} > \text{diakok}[i+1].\text{magassag})$ és $(\text{van}) \rightarrow (\text{valasz} = \text{"NEM"})$ és $(\text{nem van}) \rightarrow (\text{valasz} = \text{"IGEN"})$

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

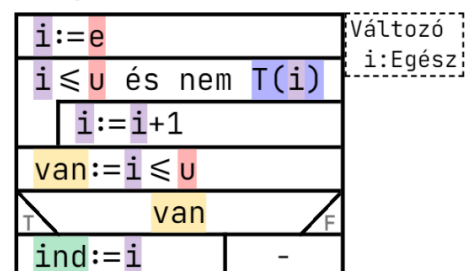
Ef: -

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



Visszavezetés

	Megszámolás	
i	~	i
e..u	~	1..t-1
T(i)	~	diakok[i].magassag > diakok[i+1].magassag
van	~	van

Algoritmus

Local declarations

t: Egész

Diak: (nev: Szöveg x magassag: Egész)

diakok: Tömb[1..t] → Diak //?

van: Logikai

valasz: Szöveg

i: Egész

Be: t, diakok

van:=hamis

i:=1

i < t és nem van

$\text{diakok}[i].\text{magassag} > \text{diakok}[i+1].\text{magassag}$

van:=igaz

van:=hamis

i:=i+1

van

valasz:="NEM"

valasz:="IGEN"

Ki: valasz

Link: Structogram editor

2. fázishoz

Kód

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

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

using System;

class Program {

    public struct Diak {
        public string nev;
        public int magassag;
    }

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

        int t;
        Diak[] diakok;
        bool van;
        string valasz;
        int i;

        #endregion

        #region Beolvasás

        t = int.Parse(Console.ReadLine());
        diakok = new Diak[t];

        for (i = 0; i < t; ++i) {
            string[] sor = Console.ReadLine().Split();
            diakok[i].nev = sor[0];
            diakok[i].magassag = int.Parse(sor[1]);
        }

        #endregion
    }
}
```

```

#region Feldolgozás

van = false;
i = 0;

while (i < t-1 && !(van)) {
    if (diakok[i].magassag > diakok[i+1].magassag) {
        van = true;
    } else {
        van = false;
    }
    i++;
}

if (van) {
    valasz = "NEM";
} else {
    valasz = "IGEN";
}

#endregion

#region Kiírás
Console.WriteLine($"{valasz}");
#endregion
}

```

Bíró eredmény

Verzió: 1.0

Total points: 100/100

Test#	Point	...Verdict...	CPU time
1.1	10/10	HELYES	0.029 sec
2.1	10/10	HELYES	0.031 sec
3.1	10/10	HELYES	0.031 sec
4.1	10/10	HELYES	0.031 sec
5.1	10/10	HELYES	0.033 sec
6.1	10/10	HELYES	0.037 sec
7.1	10/10	HELYES	0.032 sec
8.1	10/10	HELYES	0.031 sec
9.1	10/10	HELYES	0.031 sec
10.1	10/10	HELYES	0.031 sec

Date of submission: 2025-10-27 21:55:39.0