

MEGOLDÁS SABLON 1.

C. BEADANDÓ HÁZI FELADAT.

Név: Vida Levente

Neptunkód: DLNINP

Feladat a biroból (lehet képernyőkép is)

Programozási tételek: megszámlálás

*

Jelenlévők

Egy rendezvényre N vendég érkezik. Ismerjük mindegyikük érkezési és távozási idejét.

Számold meg, hogy a K . időpontban hány vendég van a rendezvényen!

Bemenet

A *standard bemenet* első sorában található a vendégek száma ($1 \leq N \leq 100$) és a K percben megadott időpillanat. Az ezt követő N sorban pedig szóközzel elválasztva az egyes vendégek érkezési és távozási ideje napon belüli percben megadva (éjfél utáni távozás esetén $24 \cdot 60$ -nál nagyobb értékek vannak).

Kimenet

A *standard kimenet* egyetlen sorába ki kell írni a K . időpontban jelen lévő résztvevők számát!

Példa

Bemenet

```
6 1440
1320 1410
1350 1430
1400 1520
1080 1480
1120 1500
1120 1410
```

Kimenet

```
3
```

Korlátok

Időlimit: 0.1 mp

Memórialimit: 32 MB

I. Részteljesítés

max: 50 pont

Specifikáció	max: 20 pont												
<p><u>Specifikáció</u></p> <p>ido = Rekord(erberez $\in \mathbb{N}$, tavozas $\in \mathbb{N}$)</p> <p>Be: $n \in \mathbb{N}$; $h \in \mathbb{N}$; $x \in \text{ido}[n]$</p> <p>Ki: $db \in \mathbb{N}$</p> <p>Ef: $1 \leq n \leq 100$, $\forall i \in [1..n]: (x[i].erberez \leq x[i].tavozas)$</p> <p>Uf: $db = \text{DARAB}(i=1..n, x[i].erberez \leq h \wedge x[i].tavozas \geq h)$</p>	<p>Itt megalkotva, vagy másolva más környezetből vagy kézzel írt megoldás fényképe</p>												
Sablon + forrásmegjelölés	max: 5 pont												
<p><u>Források</u></p> <div data-bbox="207 929 654 1254"> <p>Megszámolás sablon</p> <table border="1"> <thead> <tr> <th>i</th> <th>T(i) érték</th> </tr> </thead> <tbody> <tr> <td>e</td> <td>IGAZ 1</td> </tr> <tr> <td>e+1</td> <td>HAMIS 0</td> </tr> <tr> <td>...</td> <td>HAMIS 0</td> </tr> <tr> <td>u</td> <td>IGAZ 1</td> </tr> <tr> <td></td> <td>db=2</td> </tr> </tbody> </table> <p>Feladat</p> <p>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 háányszor veszi fel az igaz értéket!</p> <p>Specifikáció</p> <p>Be: eEZ, uEZ Ki: dbEN Ef: - Uf: $db = \text{SZUMMA}(i=e..u, 1, T(i))$ Rövidítve: Uf: $db = \text{DARAB}(i=e..u, T(i))$</p> <p>Algoritmus</p> <pre> db:=0 i:=e while i<=u if T(i) db:=db+1 i:=i+1 endwhile </pre> </div> <div data-bbox="686 929 1133 1254"> <p>Példa – téli születések visszavezetés</p> <p>Adjuk meg n születési hónap alapján, hogy közöttük hányan születtek télen!</p> <p>Feladatsablon (mintafeladat)</p> <p>Be: eEZ, uEZ Ki: dbEN Ef: - Uf: $db = \text{DARAB}(i=e..u, T(i))$</p> <p>Téli születések száma (konkrét feladat)</p> <p>Be: nEN, hóEN[1..n] Ki: dbEN Ef: $\forall i \in [1..n]: (1 \leq \text{hó}[i] \leq 12)$ Uf: $db = \text{DARAB}(i=1..n, \text{hó}[i] < 3 \text{ vagy } \text{hó}[i] = 12)$</p> <p>Visszavezetés:</p> <p>$e..u \sim 1..n$ $T(i) \sim \text{hó}[i] < 3 \text{ vagy } \text{hó}[i] = 12$</p> <p>Algoritmus:</p> <pre> db:=0 i:=e while i<=u if T(i) db:=db+1 i:=i+1 endwhile </pre> </div>	i	T(i) érték	e	IGAZ 1	e+1	HAMIS 0	...	HAMIS 0	u	IGAZ 1		db=2	<p>Lehet képernyőkép vagy másolat az előadás anyagából.</p> <p>Forrásmegjelölés : a forrásfájl neve (publikálás útvonala) és oldalszám/dia sorszáma.</p> <p>forrásfájl : eloadas03.pdf oldal : 59,60 link : http://progalap.elte.hu/downloads/eloadas/progalap_ea3.zip</p>
i	T(i) érték												
e	IGAZ 1												
e+1	HAMIS 0												
...	HAMIS 0												
u	IGAZ 1												
	db=2												

Visszavezetési táblázat (megfeleltetés)	max: 5 pont								
<h1>Visszavezetés</h1> <h2>Feladatsablon</h2> <p> $\mathcal{D}e: e \in \mathbb{Z}, u \in \mathbb{Z}$ $k_i: db \in \mathbb{N}$ $\mathcal{E}f: -$ $Uf: db = \text{DARAB}(i = e..u, T(i))$ </p> <h2>Megfeleltetés</h2> <p> $e..u \sim 1..n$ $T(i) \sim x[i].\text{erbeszer} \leq b \wedge x[i].\text{tavozas} \geq b$ </p>	<p>A specifikáció és a minta közötti kapcsolat leírása</p>								
Algoritmus	max: 20 pont								
<h1>Algoritmus</h1> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td colspan="2">$db := 0$</td> </tr> <tr> <td colspan="2">$i = 1..n$</td> </tr> <tr> <td colspan="2" style="text-align: center;">$x[i].\text{erbeszer} \leq b \wedge x[i].\text{tavozas} \geq b$</td> </tr> <tr> <td style="text-align: center;">$db := db + 1$</td> <td style="text-align: center;">—</td> </tr> </table>	$db := 0$		$i = 1..n$		$x[i].\text{erbeszer} \leq b \wedge x[i].\text{tavozas} \geq b$		$db := db + 1$	—	<p>Itt megalkotva vagy képernyőkép vagy kézzel írt megoldás fényképe</p>
$db := 0$									
$i = 1..n$									
$x[i].\text{erbeszer} \leq b \wedge x[i].\text{tavozas} \geq b$									
$db := db + 1$	—								

<h2>Specifikáció</h2> <p>ido = Rekord(erkezes $\in \mathbb{N}$, tavozas $\in \mathbb{N}$)</p> <p>Be: $n \in \mathbb{N}, b \in \mathbb{N}, x \in \text{ido}[n]$</p> <p>Ki: $db \in \mathbb{N}$</p> <p>Ef: $1 \leq n \leq 100, \forall i \in [1..n]: (x[i].erkezes \leq x[i].tavozas)$</p> <p>Uf: $db = \text{DARAB}(i=1..n, x[i].erkezes \leq b \wedge x[i].tavozas > b)$</p>	<p>Indoklás: Logikailag így helyes.</p> <p>Példa: Ha egy vendég 13:00-kor távozott a szobából, akkor 13:00-kor már nem volt a szobában.</p>								
<h2>Visszavezetés</h2> <h3>Feladatsablon</h3> <p>Be: $e \in \mathbb{Z}, u \in \mathbb{Z}$</p> <p>Ki: $db \in \mathbb{N}$</p> <p>Ef: -</p> <p>Uf: $db = \text{DARAB}(i=e..u, T(i))$</p> <h3>Megfeleltetés</h3> <p>$e..u \sim 1..n$</p> <p>$T(i) \sim x[i].erkezes \leq b \wedge x[i].tavozas > b$</p>	<p>Specifikációban történt változás miatt a visszavezetésben is változik a feltétel.</p>								
<h2>Algoritmus</h2> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="padding: 5px;">$db := 0$</td> </tr> <tr> <td colspan="2" style="padding: 5px;">$i = 1..n$</td> </tr> <tr> <td colspan="2" style="padding: 5px;"> <div style="border: 1px solid black; padding: 2px; margin: 2px;"> $x[i].erkezes \leq b \wedge x[i].tavozas > b$ </div> </td> </tr> <tr> <td style="padding: 5px; width: 50%;">$db := db + 1$</td> <td style="padding: 5px; width: 50%; text-align: center;">—</td> </tr> </table>	$db := 0$		$i = 1..n$		<div style="border: 1px solid black; padding: 2px; margin: 2px;"> $x[i].erkezes \leq b \wedge x[i].tavozas > b$ </div>		$db := db + 1$	—	<p>Specifikációban történt változás miatt az algoritmusban is változik a feltétel.</p>
$db := 0$									
$i = 1..n$									
<div style="border: 1px solid black; padding: 2px; margin: 2px;"> $x[i].erkezes \leq b \wedge x[i].tavozas > b$ </div>									
$db := db + 1$	—								

<p>biro.elte.hu téma: Programozás 1. beadandó</p> <p>feladat: Jelenlévők</p> <p>beadás (feltöltés) sorszáma: 21.</p> <p>pontszám: 90 (csak azért, mert a 2. teszt hibás)</p>
<p>Ha volt: hibás tesztesetek sorszáma és a mellettük megjelenő hibaüzenetek.</p> <p>1. teszt: Fordítási hiba</p> <p>Többi teszt: A bíróban a 2. teszt hibás, ebből kifolyólag sok „Hibás kimenet” hibaüzenetet kaptam.</p>

Kód**max 26 pont**

```
using System;
namespace elso_bead_kod
{
    internal class Program
    {
        struct ido public int erkezes, tavozas;

        static void Main(string[] args)
        {
            //Beolvasás
            int n, k;
            do
            {
                string[] fstRow = Console.ReadLine().Split(' ');
                n = int.Parse(input[0]);
                k = int.Parse(input[1]);
            } while (!(1 <= n && n <= 100));

            ido[] x = new ido[n];
            for (int i = 0; i < n; i++)
            {
                do
                {
                    string[] row = Console.ReadLine().Split(' ');
                    x[i].erkezes = int.Parse(row[0])
                    x[i].tavozas = int.Parse(row[1])
                } while (!(x[i].erkezes <= x[i].tavozas));
            }

            // Feladat:
            int db = 0;
            for (int i = 0; i < n; i++)
                if (x[i].erkezes <= k && k < x[i].tavozas)
                    db++;

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

Teszt**max 3 pont**

A feladathoz adott 2 teszten túl még 3 teszt.

	Saját 1.	Saját 2.	Saját 3
Bemenet:	6 960 900 930 910 950 930 970 920 940 940 980 950 970	6 1200 1100 1130 1120 1140 1130 1150 1150 1180 1160 1190 1140 1170	6 1080 1000 1120 1020 1040 1010 1050 1030 1070 1040 1060 1020 1090
Kimenet:	3	0	2