

Személyes adatok

Név: Barta Máté Bence

Neptun Kód: SAZZ1P

Beadás Verziószáma: 1.0

Feladat Szövege:

Idős és alacsony fizetésűek száma

Egy vállalat tárolja dolgozóinak életkorát és fizetését.

Írj programot, amely megadja, hogy az L év felettiek közül hánynak kisebb a fizetése M forintnál!

Bemenet

A *standard bemenet* első sorában a dolgozók száma ($0 \leq N \leq 100$), a korhatár ($1 \leq L \leq 100$) és a fizetési határ ($1 \leq M \leq 2\,000\,000$) van, alatta pedig soronként egy-egy ember kora ($1 \leq K \leq 100$) és a fizetése ($1 \leq F \leq 2\,000\,000$).

Kimenet

A *standard kimenet* egyetlen sorába az L év felettiek közül az M forintnál kisebb fizetésű dolgozók számát kell kiírni!

Példa

Bemenet	Kimenet
10 65 200000	4
69 180000	
68 130000	
67 100000	
66 190000	
65 140000	
64 180000	
63 100000	
68 230000	
61 190000	
60 220000	

Korlátok

Időlimit: 0.1 mp

Memórialimit: 32 MB

Pontozás: A tesztek 40%-ában a bemenet hossza ≤ 20 .

Link a Specifikációra:

[Specifikáció a feladatra](#)

A Specifikáció Szövege:

Be: $n \in \mathbb{N}$,

$l \in \mathbb{N}$,

$m \in \mathbb{N}$,

$k \in \mathbb{N}[1..n]$,

$f \in \mathbb{N}[1..n]$

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

Ef: $0 \leq n \leq 100$ és

$1 \leq l \leq 100$ és

$1 \leq m \leq 2000000$ és

$\forall i \in [1..n]: (1 \leq k[i] \leq 100)$ és

$\forall j \in [1..n]: (1 \leq f[j] \leq 2000000)$

Uf: $db = \mathbf{DARAB}(x=1..n, k[x] > l \text{ és } f[x] < m)$

Az Alkalmazott Sablon Képe:

Megszámolás sablon

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

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: $db \in \mathbb{N}$

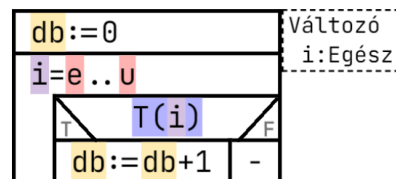
Ef: -

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

Rövidítve:

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

Algoritmus



Visszavezetési Táblázat:

n = Természetes Szám: a dolgozók száma

l = Természetes Szám: a dolgozók korhatára

m = Természetes Szám: a dolgozók fizetési határa

k = Egy Természetes Számokból álló lista, az emberek korát tartalmazza

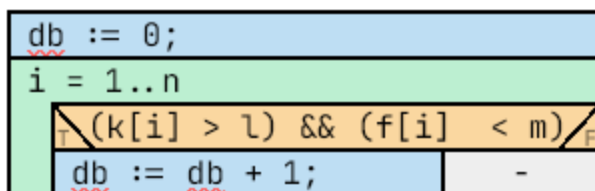
f = Egy Természetes Számokból álló lista, az emberek fizetését tartalmazza

db = Egy Természetes Szám: az összes kritériumoknak megfelelt dolgozók száma

Link az Algoritmusra:

[Az Algoritmus Linkje](#)

Algoritmus:



Kód:

```
using System;
```

```
namespace Barta_Máté_Bead_1C
```

```
{
```

```
    internal class Program
```

```
    {
```

```
        const int MaxN = 100;
```

```
        static void Main(string[] args)
```

```
        {
```

```
            /* Név: Barta Máté Bence
```

```
            * Neptun Kód: SAZZ1P
```

```
            * Email cím: bmate20050911@gmail.com <-(személyes)
```

```
            .hu bartamate@student.elte.hu
```

```
            */
```

```
            // deklarálás
```

```
            Console.Error.WriteLine("Idős és alacson fizetésűek  
száma");
```

```
            int n, l, m, db;
```

Megoldás az „1. beadandó (C1) – 2. fázis” -hoz

```
int[] k = new int[MaxN];
int[] f = new int[MaxN];

// beolvasás
Console.Error.Write("N L M? : ");
string[] tmp = Console.ReadLine().Trim().Split(" ");

n = int.Parse(tmp[0]);
l = int.Parse(tmp[1]);
m = int.Parse(tmp[2]);

Console.Error.WriteLine("Dolgozók életkora és
fizetése");
for (int i = 1; n >= i; ++i)
{
    Console.Error.Write($"{i}. dolgozó adatai : ");
    string[] tmp2 =
        Console.ReadLine().Trim().Split(" ");
    k[i - 1] = int.Parse(tmp2[0]);
    f[i - 1] = int.Parse(tmp2[1]);
}

// feldolgozás
db = 0;
for(int i = 1; n >= i; ++i)
{
    if (k[i - 1] > l && f[i - 1] < m) db++; //
    algoritmusban 1..n van írva, de mivel az array
    az 0-tól indexelődik, muszályok vagyunk kivonni
    az indexből 1-et az azonosításhoz
}

// kiíratás
Console.WriteLine(db);

}

}
```

Megoldás az „1. beadandó (C1) – 2. fázis” -hoz

Bíró pontszám:

Felhasználó: Barta Máté Bence, Téma: Programozás 1. beadandó, Határidő: 2025-10-26 23:59:59

TÉMAVÁLTÁS

BEAD

EREDMÉNY

LETÖLTÉS

VISSZATÖLT

BEADOTTAK

FŐOLDAL

KILÉP/BELÉP

Utolsó beadás eredménye

Összpont: 100/

Teszt#	Pont	...Verdikt...	futási idő
1.1	3/3	Helyes	0.034 sec
2.1	3/3	Helyes	0.032 sec
3.1	3/3	Helyes	0.031 sec
4.1	3/3	Helyes	0.033 sec
5.1	3/3	Helyes	0.033 sec
6.1	3/3	Helyes	0.031 sec
7.1	3/3	Helyes	0.034 sec
8.1	3/3	Helyes	0.033 sec
9.1	4/4	Helyes	0.032 sec
10.1	4/4	Helyes	0.033 sec
11.1	4/4	Helyes	0.032 sec
12.1	4/4	Helyes	0.032 sec
13.1	4/4	Helyes	0.033 sec
14.1	4/4	Helyes	0.033 sec
15.1	4/4	Helyes	0.033 sec
16.1	4/4	Helyes	0.033 sec
17.1	4/4	Helyes	0.032 sec
18.1	4/4	Helyes	0.033 sec
19.1	4/4	Helyes	0.033 sec
20.1	4/4	Helyes	0.033 sec
21.1	4/4	Helyes	0.038 sec
22.1	4/4	Helyes	0.032 sec
23.1	4/4	Helyes	0.035 sec
24.1	4/4	Helyes	0.033 sec
25.1	4/4	Helyes	0.032 sec
26.1	4/4	Helyes	0.032 sec
27.1	4/4	Helyes	0.032 sec

Beadva: 2025-10-20 10:50:12.0

Saját tesztfájlok:

be3.txt: (a tesztfileban nyilvánvalóan egymás alatt vanna kezek a sorok, szimplán helytakarékosság miatt rendeztem őket 3 oszlopba)

80 55 500000	30 358000	56 19000
63 870000	26 750000	21 441000
68 473000	35 607000	34 979000
59 800000	38 325000	26 359000
43 520000	48 38000	34 480000
59 678000	41 634000	25 688000
25 720000	69 958000	35 880000
51 582000	25 652000	40 918000
27 537000	30 635000	23 216000
67 758000	28 995000	54 565000
46 105000	52 581000	48 865000
40 473000	32 414000	33 508000
33 186000	43 474000	46 916000
58 736000	32 623000	24 921000
42 216000	27 338000	48 83000
48 135000	25 674000	66 277000
20 324000	52 317000	35 9000
50 149000	26 778000	53 842000
50 222000	29 949000	26 647000
50 386000	38 662000	55 841000
67 902000	61 13000	34 264000
54 449000	24 622000	29 397000
37 613000	61 673000	49 552000
41 902000	24 971000	21 164000
54 99000	68 878000	61 369000
23 969000	43 509000	20 146000
53 653000	68 55000	Eredmény: 6
53 170000	50 451000	

Megoldás az „1. beadandó (C1) – 2. fázis” -hoz

be4.txt:

70 51 580000	69 958000	26 359000
46 105000	25 652000	34 480000
40 473000	30 635000	25 688000
33 186000	28 995000	35 880000
58 736000	52 581000	40 918000
42 216000	32 414000	23 216000
48 135000	43 474000	54 565000
20 324000	32 623000	48 865000
50 149000	27 338000	33 508000
50 222000	25 674000	46 916000
50 386000	52 317000	24 921000
67 902000	26 778000	48 83000
54 449000	29 949000	66 277000
37 613000	38 662000	35 9000
41 902000	61 13000	53 842000
54 99000	24 622000	26 647000
23 969000	61 673000	55 841000
53 653000	24 971000	34 264000
53 170000	68 878000	29 397000
30 358000	43 509000	49 552000
26 750000	68 55000	21 164000
35 607000	50 451000	61 369000
38 325000	56 19000	20 146000
48 38000	21 441000	Eredmény: 9
41 634000	34 979000	

Megoldás az „1. beadandó (C1) – 2. fázis” -hoz

be5.txt:

90 48 620000	58 736000	26 778000
47 592000	42 216000	29 949000
55 844000	48 135000	38 662000
50 857000	20 324000	61 13000
47 847000	50 149000	24 622000
41 623000	50 222000	61 673000
52 384000	50 386000	24 971000
41 297000	67 902000	68 878000
64 56000	54 449000	43 509000
68 272000	37 613000	68 55000
39 477000	41 902000	50 451000
59 812000	54 99000	56 19000
46 479000	23 969000	21 441000
48 392000	53 653000	34 979000
66 836000	53 170000	26 359000
23 337000	30 358000	34 480000
24 648000	26 750000	25 688000
21 368000	35 607000	35 880000
61 957000	38 325000	40 918000
58 140000	48 38000	23 216000
63 870000	41 634000	54 565000
68 473000	69 958000	48 865000
59 800000	25 652000	33 508000
43 520000	30 635000	46 916000
59 678000	28 995000	24 921000
25 720000	52 581000	48 83000
51 582000	32 414000	66 277000
27 537000	43 474000	Eredmény: 20
67 758000	32 623000	
46 105000	27 338000	
40 473000	25 674000	
33 186000	52 317000	