

”Programozás”
beadandó feladat:
4. feladat

Készítette: Bárdosi Bence
Neptun-azonosító: VY9NJN
E-mail: bardosi.bence@gmail.com

2017-03-06

Tartalom

1	Dokumentáció	2
1.1	Feladat	2
1.2	Specifikáció	2
1.3	Algoritmus	2
1.4	Implementáció	2
1.4.1	Adattípusok megvalósítása	2
1.4.2	Bemenő adatok formája	2
1.4.3	Függvények kapcsolódási szerkezete	3
1.5	Tesztelés	3
1.5.1	A feladat specifikációjára épülő (fekete doboz) tesztesetek:	3
1.5.2	A megoldó programra épülő (fehér doboz) tesztesetek:	3

1 Dokumentáció

1.1 Feladat

Egymást követő napokon megmértük a napi hőmérsékletet, és ezeket az értékeket egy tömbben tároljuk. Hányszor mértünk 0 Celsius fokot úgy, hogy másnap fagypont alatti hőmérséklet volt?

1.2 Specifikáció

$$\mathbf{A} = (t : \mathbb{R}^n, c : \mathbb{N})$$

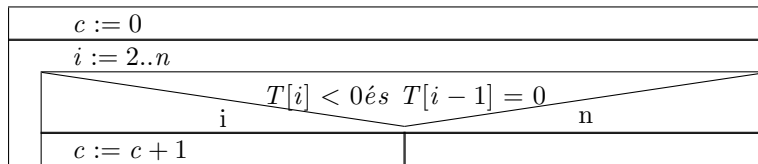
$$\mathbf{E}f = (t = t')$$

$$\mathbf{U}f = \left(Ef \wedge c = \sum_{\substack{i=2 \\ t[i] < 0 \wedge t[i-1] = 0}}^n 1 \right)$$

1.3 Algoritmus

A feladatot az összegzés programozási tételére vezetjük vissza.

Tétel	Feladat
$m \leftarrow$	2
$n \leftarrow$	n
$\beta(i) \leftarrow$	$t[i] < 0 \wedge t[i-1] = 0$



1.4 Implementáció

1.4.1 Adattípusok megvalósítása

A kódoláskor a `t` tömböt `vector<float>`-ként deklaráljuk, amelynek mérete `t.size()` alakban érhető el. Mivel a vektor 0-tól indexelődik, azért a tervbeli ciklus nem a `2..n`, hanem az `1..n-1` intervallumot, pontosabban a `1..t.size()-1` intervallumot futja be, aminek következtében a struktogramm kódja az alábbi lesz:

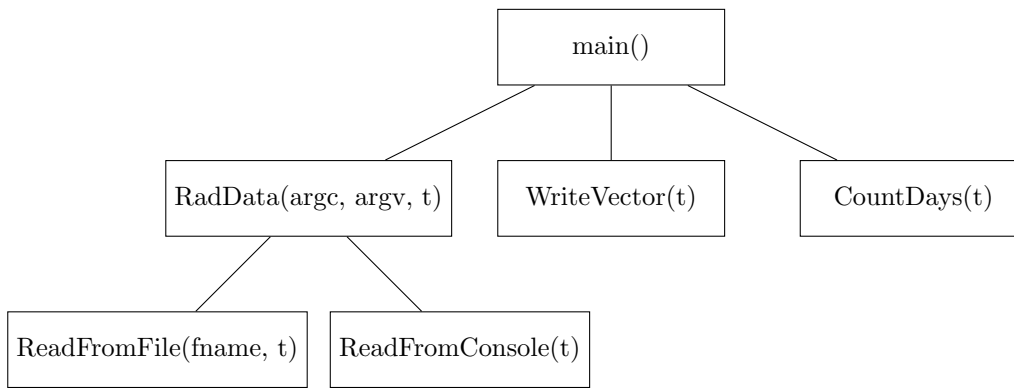
```

1      int c=0;
2      for(int i=1; i<(int)t.size(); ++i)
3          if(t[i]<0 && t[i-1]==0)
4              ++c;}
```

1.4.2 Bemenő adatok formája

A bemenő adatokat egy szöveges állományból kell a tömbbe bemásolni. Az állományban a megadott értékeket szóközzel, tabulátor jelekkel vagy sorvége jelekkel elválasztva kell beírni. Az állomány minden sorát sorvége jel zárja le.

1.4.3 Függvények kapcsolódási szerkezete



1.5 Tesztelés

1.5.1 A feladat specifikációjára épülő (fekete doboz) tesztesetek:

Megszámolás tétel tesztesetei:

intervallum hossza szerint:

- | | |
|-------------------------|---|
| 1. <i>nulla</i> hosszú: | Egyetlen nap sincs
be1.txt: [] - válasz: 0 |
| 2. <i>egy</i> hosszú: | Egyetlen nap
be2.txt: [5.2] - válasz: 0 |
| 3. <i>kettő</i> hosszú: | Kettő, a feltételnek eleget nem tevő nap
be3.txt: [5.2, 0] - válasz: 0
Kettő, a feltételnek eleget tevő nap
be3.txt: [0, -2] - válasz: 1 |
| 4. <i>több</i> hosszú: | Több nap
be5.txt: [0, -2, 0, 5, 6, 0, -0.5] - válasz: 2 |

intervallum eleje szerint:

Sorozat elején található csak a feltételnek megfelelő nappár
be6.txt: [0, -2, 0, 5, 6, 0] - válasz: 1

intervallum vége szerint:

Sorozat végén található csak a feltételnek megfelelő nappár
be7.txt: [0, 2, 0, 5, 6, 0, -2] - válasz: 1

tételre jellemző esetek szerint:

- | | |
|--|---|
| 1. Egyetlen megfelelő nappár van: | be7.txt: [0, 2, 0, 5, 6, 0, -2] - válasz: 1 |
| 2. Nincs megfelelő nappár: | be8.txt: [0, 1, 2, 3, 4] - válasz: 0 |
| 3. Egy "0" napot több "negatív" nap követ: | be9.txt: [1, 2, 0, -1, -2, -1] - válasz: 0 |

1.5.2 A megoldó programra épülő (fehér doboz) tesztesetek:

- Hibás vagy nem létező állománynév megadása.
- Állomány nevének megadása parancssorból.
- Olyan állomány olvasása, ahol egy sorban több érték is található egyetlen illetve több szóközzel és/vagy tabulátor jellel elválasztva (be10.txt).
- Olyan állomány olvasása, ahol minden érték külön sorban van (be9.txt).
- Olyan állomány olvasása, ahol az utolsó sort nem zárja sorvége jel, és éppen ennek a sornak a tartalma határozza meg az eredményt (adat: [1, 2, 3, 0, -1] – válasz: 1) (be11.txt).
- Főprogram ciklusának ellenőrzése: olyan bemenő adatokkal, amelyekre a ciklus egyszer sem fut le (Pl: be1.txt), pontosan egyszer fut le (Pl: be3.txt), vagy többször lefut (Pl: be5.txt).