
Nombre de subseqüències felices i tristes

X68437_ca

En aquest exercici heu de fer varies coses.

En primer lloc, haureu d'implementar una funció que, donat un string s i tres caràcters diferents c_1, c_2, c_3 rebuts com a paràmetre, retorna quants cops apareix $c_1c_2c_3$ com a subseqüència (no-consecutiva) dins de s . En altres paraules, retorna el nombre de tripletes de índexos (i_1, i_2, i_3) tals que $i_1 < i_2 < i_3$ i $s[i_1] = c_1$, $s[i_2] = c_2$, $s[i_3] = c_3$. Aquesta és la capcelera:

```
// Pre: c1, c2, c3 are pairwise different characters.
// Post: returns the number of triples (i1, i2, i3) such that 0<=i1<i2<i3<s.size(
//       s[i1]=c1, s[i2]=c2, s[i3]=c3.
int numberSubsequences(const string &s, char c1, char c2, char c3);
```

Nota: els jocs de proves privats d'aquest exercici son grans i estan dissenyats per a que calgui una implementació de cost lineal de `numberSubsequences`. Una implementació lenta us permetrà només superar els jocs de proves públics i obtenir la meitat de la nota.

En segon lloc, haureu d'implementar dues funcions que calculen el nombre de subseqüències felices i tristes en un string, respectivament. Una subseqüència feliç és una subseqüència de tres caràcters, i a on aquests tres caràcters són, o bé `:-)` o bé `(-:`, en l'ordre donat. Una subseqüència trista és una subseqüència de tres caràcters, i a on aquests tres caràcters són, o bé `:(` o bé `)-`, en l'ordre donat. Aquestes son les capceleres:

```
// Pre:
// Post: returns the number of triples (i1, i2, i3) such that 0<=i1<i2<i3<s.size(
//       either s[i1]=':', s[i2]='-', s[i3]=')' or s[i1]='(', s[i2]='-', s[i3]='
int numberHappySubsequences(const string &s);
```

```
// Pre:
// Post: returns the number of triples (i1, i2, i3) such that 0<=i1<i2<i3<s.size(
//       either s[i1]=':', s[i2]='-', s[i3]='(' or s[i1]=')', s[i2]='-', s[i3]='
int numberSadSubsequences(const string &s);
```

Les dues funcions anteriors hauran d'usar convenientment la funció `numberSubsequences` mencionada al principi. En cas contrari, s'invalidarà l'entrega.

Finalment, heu d'implementar un programa principal que llegeix strings d'entrada, i per a cadascun d'ells escriu el seu nombre de subseqüències felices i el seu nombre de subseqüències tristes.

Aquest programa haurà d'usar convenientment les funcions mencionades anteriorment. En cas contrari, s'invalidarà l'entrega.

Entrada

L'entrada té varis strings sobre `{ ':', '- ', '(,)' }`, cadascun en una línia.

Sortida

Per a cada cas, la sortida té dos nombres separats per un espai en blanc en una línia, el nombre de subseqüències felices i el nombre de subseqüències tristes.

Exemple d'entrada

```
-) :)
-() -:-: :-(-((:(
)---:::)
--)
) (: -)) :
)- (: -
-: (( -- ()) ) (: (
-)) ) (:
)-:
(:-: ((-((:-:-(
(: (---::---:)) -) (
() (-)) ) (: ) :
): ) : : ( ) -)-: : ((: (
:-(( ( ) ) (:
)-: ) : : (( ( ) -) (-((-: (
::) :
(:) (: :-)-:)) ) - (:
() ((: ) -- (-
())) ((:-:
: ( ) (-) : --) -
```

Exemple de sortida

```
0 0
10 43
0 9
0 0
4 1
0 1
10 8
0 0
0 1
18 22
64 16
4 2
10 51
2 8
16 39
0 0
35 25
0 8
3 3
8 1
```

Observació

Avaluació sobre 10 punts:

- Solució lenta: 5 punts.
- solució ràpida: 10 punts.

Entenem com a solució ràpida una que és correcta, de cost lineal i capaç de superar els jocs de proves públics i privats. Entenem com a solució lenta una que no és ràpida, però és correcta i capaç de superar els jocs de proves públics.

Informació del problema

Autor : PRO1

Generació : 2023-11-27 23:07:04

© *Jutge.org*, 2006–2023.

<https://jutge.org>