Nombre de substrings que son ben parentitzats X87610_ca

Preliminars:
En aquests preliminars expliquem qué és un mot ben-parentitzat sobre `(,)`. Si ja teniu clar aquest concepte, podeu deixar de llegir els preliminars i anar directament a l'exercici en sí.
Un mot ben parentitzat és un string `s` format amb els caràcters d'obrir i tancar parèntesis, és a dir `(,)`, que cumpleix el següent. A base de reemplaçar la subseqüència `()` pel mot buit, tant com sigui possible, *s* s'acaba convertint en el mot buit.
Per exemple, considereu el mot `(()(()()))`. Si reemplacem les tres ocurrències de `()` pel mot buit ens queda `(())`. Ara, si reemplacem la ocurrència de `()` pel mot buit ens queda `()`. Finalment, si reemplacem la nova ocurrència de `()` pel mot buit ens queda el mot buit. Per tant, el mot inicial era ben parentitzat.
Considereu aquest altre exemple: `())(()`. Si reemplacem les dues ocurrències de `()` per mots buits ens queda `)(`. Aquí ja no podem aplicar cap més reemplaçament. Per tant, el mot inicial no era ben-parentitzat.
Exercici:
Tindrem strings d'entrada que son ben-parentitzats. Heu d'implementar un programa que, per a cada cas, indica quantes ocurrències de substrings seus no-buits son parentitzats.
Per exemple, amb el cas d'entrada `(()(()()))`, tenim 3 ocurrències del substring ben-parentitzat no-buit `()`, 1 ocurrència del substring ben-parentitzat no-buit `(()())`, i 1 ocurrència del substring ben-parentitzat no-buit `(()(()()))`, i no hi ha cap altra ocurrència de substring ben-parentitzat no-buit. Per tant, amb aquest cas d'entrada, la resposta és 3+1+1, és a dir 5.
Observació: Podeu seguir l'enfocament que considereu oportú, i podeu utilitzar qualsevol de les estructures de dades presentades al curs (**string, vector, stack, queue, list, map, set**) de la manera que considereu oportuna. Noteu, però, que enfocaments diferents poden donar lloc a solucions més o menys eficients, i que superin només els jocs de proves públics o tots els jocs de proves, de manera que la nota acabarà depenent d'això.

Entrada

L'entrada conté un nombre arbitrari de casos, un per línia. Cada cas consisteix en un string ben-parentitzat no-buit sobre `(,)`.

Sortida

Per a cada cas, escriviu en una línia el nombre d'ocurrències de substrings ben-parentitzats no-buits de l'string d'entrada.

Exemple d'entrada 1

```
()
(())
()()
(()())
((()))
(())(())
((()))()
()((()))
```

```
()()()()
(()())(()())
(((()))((()))((())))
(((())())(()())))
(((())())(()()))((()())(()()))
()(()(()()))(((((()())))))(((()())()))()(()())
```

Exemple de sortida 1

```
1
2
3
4
3
5
5
5
10
9
12
10
21
45
```

Exemple d'entrada 2

```
( ( ( () () ) () ) )
( () ) ( ( () ( ( () ) () ) ) )
( ( () ) )
( () )
() ( ( ( ( ( ( ( ( () ) ) ) ) ) ) ) )
() () ( ( ( ( () ) ) ) () ( () ) () )
( () ( ( ( ( ( () ) ) ) ) ) )
()
( () ) () ( ( () ) )
()
( ( ( ( () ) ) () ) )
( () () ( () ) )
() ()
()
() () ( () )
() ( () () )
( () ( () ) )
( ( ( () ( () () ) ) ) )
( () ) () ( () )
( () )
( ( ( () ) ( ( () () () ) ) ) () () () () () )
( () )
( () ( () ( ( ( () () ) ( () ) ) ) ) )
( ( ( ( ( ( ( () ) ) ) ) ) ) () )
( ( () () ) )
( ( () () ) )
( ( () ( ( () ) () ( () ( () () () ) () ) () () ) ) )
( ( ( ( () ) ) ) )
()
() ( ( ( ( ( ( ( () ) ) ) ) ) ) )
()
() () ( ( ( ( ( ( ( () ) () () ) ) ) ) ) () )
() () ()
( ( () ) )
( ( () ) ) () ( () )
( ( ( () ( () ) () ) ) )
( ( ( ( () ) ) () ) () )
( ( ( ( () () ) ) ) () ( () ) )
( () )
( ( ( ( ( () ) ) ) ) )
```

Exemple de sortida 2

```
8
12
3
2
11
18
9
1
9
1
7
8
3
1
7
6
5
9
8
2
33
2
15
10
5
5
33
5
1
10
1
20
6
3
9
10
9
13
2
6
```

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 : PRO2
Generació : 2024-06-30 22:09:52