

---

## Indexar seqüències ben parentitzades

X80203\_ca

---

### Preliminars:

En aquests preliminars expliquem qu  s   s una seq  ncia ben-parentitzada sobre (,), i quin   s el corresponent par  ntesis de tancar per a cada par  ntesis d'obrir. Si ja teniu clars aquests conceptes, podeu deixar de llegir els preliminars i anar directament a l'exercici en s  .

Una seq  ncia ben parentitzada   s un string  $s$  format amb els car  cters d'obrir i tancar par  ntesis,   s a dir ( i ), que compleix les seg  ents condicions:

- Tot prefix de  $s$  t   m  s o igual par  ntesis d'obrir que de tancar.
- $s$  t   en total la mateixa quantitat de par  ntesis d'obrir que de tancar.

Sigui  $s$  una seq  ncia ben parentitzada i sigui  $i$  una posici   de  $s$  a on hi trobem un par  ntesi d'obrir (  s a dir  $s[i] == '('$ ). Sigui  $j$  la posici   m  s petita d'entre les que compleixen  $i < j$  i tals que el substring  $s[i..j]$  t   tants par  ntesis d'obrir com de tancar. Resulta que a posici    $j$  hi ha d'haver un par  ntesis de tancar, i diem que aquest   s el corresponent par  ntesis de tancar al par  ntesis d'obrir que es troba a posici    $i$ .

### Exercici:

Escriviu un programa que rep seq  ncies ben-parentitzades d'entrada i les torna a escriure per la sortida, per   insertant un n  mero darrera de cada par  ntesi de manera que:

- El primer par  ntesi d'obrir est   seguit d'un 1, el segon par  ntesi d'obrir d'un 2, el tercer par  ntesi d'obrir d'un 3, i aix   successivament.
- Per a cada par  ntesis d'obrir, el seu corresponent par  ntesis de tancar est   seguit del mateix n  mero.

**Observaci  :** Conv   que utilitzeu la classe `stack` per a resoldre aquest exercici de manera eficient.

### Entrada

L'entrada cont   un nombre arbitrari de casos, un per l  nia. Cada cas consisteix en un string ben parentitzat.

### Sortida

Per a cada cas, escriviu en una l  nia el mateix string, per   afegint darrera de cada par  ntesis un n  mero, de manera que els par  ntesis d'obrir estan identificats comen  ant des de 1 i creixentment de un en un, i els seus corresponents par  ntesis de tancar estan identificats pels mateixos n  meros.

### Exemple d'entrada 1

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

```
| (( )) (( ))  
| () () () () (( )) (( ))  
| () () () () (( )) (( ))
```

## Exemple de sortida 1

```
(1) 1
(1) 1 (2) 2
(1) 1 (2) 2 (3) 3
```

## Exemple d'entrada 2

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

```
(1 (2) 2) 1
(1 (2 (3) 3) 2) 1
(1 (2) 2 (3) 3) 1 (4 (5) 5 (6) 6) 4
(1) 1 (2 (3) 3) 2 (4 (5) 5 (6) 6) 4 (7 (8 (9) 9 (10) 10) 8 (11 (12) 12 (13) 13) 7
(1 (2) 2 (3 (4) 4) 3 (5 (6) 6 (7) 7) 5 (8 (9 (10) 10 (11) 11) 9 (12 (13) 13) 8)
```

## Exemple de sortida 2

```
(1) 1 (2) 2
(1 (2) 2 (3 (4) 4 (5) 5) 3 (6) 6) 1
(1 (2) 2) 1 (3) 3
(1) 1 (2) 2 (3 (4) 4) 3 (5 (6) 6) 5 (7) 7
(1 (2 (3) 3 (4) 4) 2) 1 (5 (6) 6) 5
(1 (2 (3) 3 (4) 4) 2 (5 (6) 6) 5) 1
(1) 1 (2) 2
(1) 1 (2 (3) 3) 2
(1) 1
(1 (2) 2 (3) 3) 1 (4) 4
(1 (2 (3) 3) 2) 1 (4) 4
(1 (2) 2 (3) 3) 1
(1 (2) 2) 1 (3) 3 (4) 4
(1 (2) 2) 1 (3 (4) 4) 3
(1) 1
(1 (2 (3) 3) 2) 1 (4) 4
(1 (2 (3 (4 (5) 5) 4) 3) 2) 1
(1 (2) 2 (3) 3) 1 (4) 4 (5 (6) 6) 5
(1) 1 (2) 2
(1) 1 (2 (3) 3 (4 (5) 5) 4 (6) 6) 2
(1) 1
(1) 1
(1) 1
(1 (2 (3 (4) 4) 3) 2 (5) 5 (6) 6) 1
(1) 1 (2 (3 (4) 4) 3) 2 (5) 5
(1 (2) 2) 1 (3 (4 (5 (6) 6) 5 (7) 7) 4) 3
(1 (2) 2 (3 (4) 4) 3) 1
(1 (2) 2) 1 (3) 3 (4) 4
(1 (2) 2) 1
(1 (2 (3 (4 (5) 5) 4) 3) 2) 1 (6) 6
(1) 1 (2 (3) 3) 2 (4) 4
(1) 1 (2 (3) 3) 2
(1 (2 (3) 3) 2) 1 (4 (5 (6) 6) 5) 4
(1 (2 (3) 3) 2) 1
(1) 1
(1 (2) 2 (3 (4) 4) 3 (5) 5) 1 (6) 6
(1) 1
(1 (2) 2) 1
(1 (2 (3) 3) 2) 1
(1 (2) 2 (3) 3) 1
```

## Observació

### Informació del problema

Autor : PRO2

Generació : 2023-09-22 01:15:22

© *Jutge.org*, 2006–2023.

<https://jutge.org>