

---

## Canviar paréntesis i corxets de tancar per a produir una seqüència ben parentitzada

---

X62454\_ca

### Preliminars:

En aquests preliminars expliquem què 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 paréntesis o corxets, és a dir `(, )`, `[, ]`, que compleix el següent. A base de reemplaçar la subseqüència `()` pel mot buit, i la subseqüència `[]` pel mot buit, tant com sigui possible, `s` s'acaba convertint en el mot buit.

Per exemple, considereu el mot `(( [ () [] ] )`. Si reemplaçem les dues ocurrences de `()` pel mot buit ens queda `[ [] ]`. Ara, si reemplaçem la ocurrència de `[]` pel mot buit ens queda `( [ ] )`. Ara, si reemplaçem la nova ocurrència de `[]` pel mot buit ens queda `()`. Finalment, si reemplaçem 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 reemplaçem la ocurrència de `()` i la ocurrència 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 poden no ser ben-parentitzats, però que segur que es poden convertir en ben-parentitzats a base de canviar alguns `)` per `]`, i canviar alguns `]` per `)`. Heu d'implementar un programa que, per a cada cas, indica quants canvis s'han de fer per tal que es converteixi en ben-parentitzat.

**Observació:** Podeu seguir l'enfoc que considereu oportú, i podeu utilitzar qualsevol de les estructures de dades presentades al curs (**string**, **vector**, **stack**, **queue**, **list**, **map**) 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 no buit sobre `(, )`, `[, ]` tal que, canviant alguns `)` per `]` i canviant alguns `]` per `)` es pot convertir en ben-parentitzat. Pot ser el cas que l'string ja sigui ben-parentitzat i que per tant calguin 0 canvis.

### Sortida

Per a cada cas, escriviu en una línia el nombre de canvis que calen per a convertir l'string en ben parentitzat.

### Exemple d'entrada 1

<code>()</code>	<code>( [ ] )</code>
<code>[]</code>	<code>[ ( ] )</code>
<code>()</code>	<code>( [ ] )</code>
<code>[]</code>	<code>[ ( ) ]</code>
<code>()</code>	<code>( ) [ ] ( [ ] )</code>
<code>[]</code>	<code>( ( ) ) [ ( ) ( ) ]</code>

```

((() ([]) [] ) [() ]
([() ([]) [] ) [() ]
[] [() ) ( () )] [ ( () ) ) [ ( () ) ) )
[ ( [] ( [] ) ) [] ] [ ( [] [ ( [] ( [] ) ) [] ] [ ( () )
( () ( () ) ( () ) ) ( ( () ( () ) ( () ( () ) ) )
[ [] ( () ) ( () ( () ) ( ( () ( () ) ( () ( () ) ] ]
[ [] [ [] ] [ [] [ [] ] [ [] [ [] ] [ [] [ [] ] ] ]

```

## Exemple de sortida 1

```

1
1
0
0
2
2
0
0
2
2
4
0
5
4
0
2
0

```

## Exemple d'entrada 2

```

(( ( ( ) ) ) ( )
[ [] [ [] ] ) ( ( [ [] ] ) ( ) [ ( ) )
[ [ ( ) ] ] ( ) [ [] ]
[ ( ( ( [ ] ) ) ) [ [ [] ] )
[ ]
[ [ [ [] ] ( ) [ ] ( ) ) [ ( [ [] ] ) ]
[ ( ( ) ) ( ( [ ] ) ) [ [ [] ] )
( ( ( ) ) ) ( )
[ ]
[ ]
[ ( ) ] [ [ [] ] [ [] ] [ ( [ [] ] ) ]
( ( ( ( ) ) ) ( [ ] ) ( ) ( ) [ ]
[ ]
( [ ] ( ) [ [ [ [] ] [ ] ] ) ( [ ] [ ]
[ [] [ ]
( )
( [ [] ] [ ] ( ) [ [] ] [ ]
[ ] ( ( ( [ [] ] ) ) )
( [ [] ] [ [] ]
[ [] [ ] ( [ [] ] [ ]
[ ( ( [ ] ( ( ) ) ) [ ( [ [ [] ] ( ) ) ] ]
[ [] [ ]
[ [ [ [] ] [ [] ] ( ( ) )
( [ [] [ ( ) ] ( ) ]
[ [ [] ] [ [] ] [ ]
[ [ ] ( ) [ [ [ [] ] ] ) ( ) ( )
( ( ) ) ( ) ( )
[ ]
( [ ( ) ( ) ] ) ( )
[ ( ) ]
( () ( ) ) [ ( [ [] ] [ [ [] ] ) ( )
[ ( ( [ ] ) )
( [ [] ( ) ) [ [ [] ] ]
[ [ [ [] ] ( [ ( ) ] ) ] [ ]
( )
[ [ [ [] ] [ [] ] ( ) ( ) )
( ( ( ( ) ) ) ( [ [] ] [ ( [ [] ] ) ) )
( [ ( [ [] ] ( ) ) ( ) ( ) ( ) ) [ [ [] ] )
[ [] ( )
[ [ ] ( [ ] ) ( )

```

## Exemple de sortida 2

```

0
4
0
4
1
2
1
0
0
0
3
1
1
3
0
0
4
1
1
2
4
0
0
1
2
4
0
1
0
0
3
2
1
2
0
2
2
2
0
4

```

## 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

Autoria: PRO2

Generació: 2026-01-25T17:21:00.164Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>