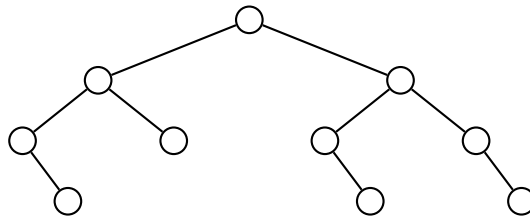


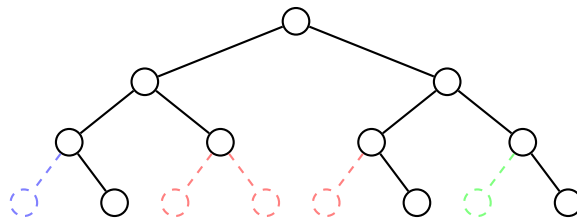
## Árboles binarios mellados



Suponemos un árbol no vacío de altura  $k \geq 1$  cuyos niveles están completos hasta la altura  $k-1$ . Es decir, para todo  $i$  tal que  $1 \leq i \leq k-1$ , el nivel  $i$ -ésimo del árbol tiene  $2^{i-1}$  nodos. El nivel  $k$ -ésimo puede estar completo o no. Si no lo está, llamamos *mella* al "hueco" que dejan las hojas consecutivas que faltan en el último nivel si lo recorremos de izquierda a derecha. Por ejemplo, suponiendo el siguiente árbol:

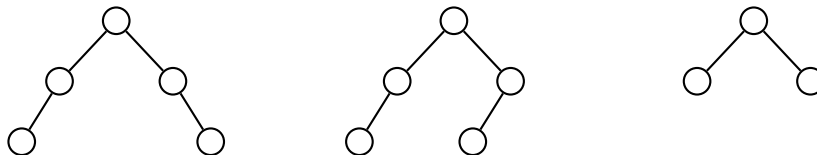


El árbol tiene altura 4 y sus niveles 1º, 2º y 3º están completos. Si completásemos el último nivel, pintando del mismo color las hojas consecutivas, obtendríamos el siguiente árbol:



Por tanto, decimos que el árbol original tiene **tres** mellas. La primera corresponde a las hojas de color azul del árbol completado, la segunda corresponde a las hojas de color rojo, y la tercera a las hojas de color verde.

A continuación se muestran más ejemplos:



El árbol de la izquierda tiene una mella. El árbol del centro tiene dos mellas. El árbol de la derecha no tiene mellas, porque tiene altura 2 y su último nivel (el 2º) está completo.

Se pide:

1. Implementar una función `num_mellas` con la siguiente cabecera:

```
template <typename T>  
int num_mellas(const BinTree<T> &t)
```

La función debe devolver el número de mellas del árbol  $t$  pasado como parámetro. Puedes suponer que el árbol no es vacío y que todos sus niveles están completos, salvo el nivel inferior, que puede estar completo o no.

Puedes implementar las funciones auxiliares que sean necesarias.

2. Indica y justifica el coste de la función `num_mellas` y de las funciones auxiliares que hayas implementado.

## Entrada

La entrada comienza con un número que indica el número de casos de prueba que vienen a continuación. Cada caso de prueba consiste en una línea con la descripción de un árbol binario mediante la notación vista en clase. El árbol vacío se representa mediante `.` y el árbol no vacío mediante `(iz x dr)`, siendo  $x$  la raíz,  $e$   $iz$  y  $dr$  las representaciones de ambos hijos.

La entrada no contiene árboles vacíos.

## Salida

Para cada caso de prueba debe imprimirse una línea con el número de mellas del árbol.

### Entrada de ejemplo

```
4
(((. * .) * .) * (. * (. * .)))
(((. * .) * .) * ((. * .) * .))
((. * .) * (. * .))
(((. * (. * .)) * (. * .)) * ((. * (. * .)) * (. * (. * .))))
```

### Salida de ejemplo

```
1
2
0
3
```

## Créditos

**Autor:** Manuel Montenegro