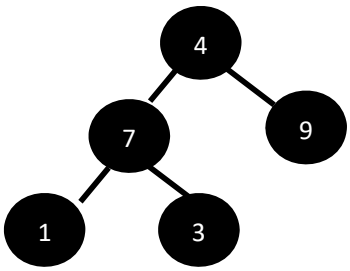
Sum of Deep Nodes

Implement a program in C++ that receives input from a binary tree “t” and a depth threshold “d”, and provides sum of the content of all nodes whose depth is greater than the depth threshold.

For example, for the following tree:



the output for d=0 will be 24 (i.e. 4+7+9+1+3).

The output for d=1 is 20 (i.e. 7+9+1+3).

The output for d=2 is 4 (i.e. 1+3).

The output for d=3 is 0.

Input

The first line will indicate the number of cases. Each case will be defined with two lines, which respectively include the binary tree of integers and the depth threshold.

Each binary tree is represented with a string recursively, in which:

- # represents an empty tree
- [n] represents a tree with just one element on the root with the n number
- (left n right) represents a tree with “n” number as root element, the tree represented by “left” recursively and the tree represented by “right” recursively.

The presented binary tree example is represented as follows with this notation:

(([1] 7 [3]) 4 [9])

Output

The output of each case should be printed in one line. The output of each case will be the sum of the deep nodes.

Implementation Details

In the virtual campus, there is some supporting material for helping you in reading binary trees from the standard input.

Example of input

```
7
(([1] 7 [3]) 4 [9])
0
(([1] 7 [3]) 4 [9])
1
(([1] 7 [3]) 4 [9])
2
(([1] 7 [3]) 4 [9])
3
(([1] 7 [3]) 4 #)
1
(# 4 ([1] 7 [6]))
1
#
0
```

Example of output

```
24
20
4
0
11
14
0
```