**8.6 Assignment 6, due Thursday 2018-04-05, at 11:00pm**

For this assignment, you are to read in a description of an ordered tree from stdin, reorder the children of each node by descreasing order of the size of their subtrees, and write the result to stdout.

A **tree** in this context consists of a **root** node and 0 or more subtrees. Each subtree also consists of a root node and 0 or more subtrees. The subtrees below each node are ordered. This means that the trees crudely depicted below are not identical, even though they have the same subtrees in a different order:

\* \*

/|\ /|\

\* \* \* \* \* \*

| |\ /| |

\* \* \* \* \* \*

The **size** of a subtree is the number of nodes in the subtree. The smallest possible subtree consists of a single root node with no children and has size 1. Both of the trees in the picture above have size 7, with the three subtrees of the root having sizes 1, 2, and 3 in the tree on the left and 3, 2, and 1 in the tree on the right.

When sorting by decreasing size, you should reorder the subtrees below each node so that the subtree with largest size comes first, followed by next largest, and so on until you reach the smallest subtree. You may break ties between subtrees of equal size arbitrarily.

8.6.1 Text representation of a tree

For the input and output to your program, a tree will be represented recursively as a pair of square brackets with all subtrees of the root represented in order between them. Some examples of representations and the corresponding trees are given below:

[] [[]] [[][]] [[][[]]] [[][[]][[][]]]

\* \* \* \* \*

| / \ / \ /|\

\* \* \* \* \* \* \* \*

| | |\

\* \* \* \*

Your program should read from stdin until it obtains the complete representation of a tree. You do not need to check for extra characters that come after the tree (so []! is not an error), but if you get an incomplete tree ([), mismatched brackets (]), or extraneous characters that are not brackets ([cow]), your program should produce no output and exit with a nonzero exit code.

The output should match the format of the input. You should not write anything other than the representation of the tree in the non-error case.

8.6.2 Examples

Here are the sorted versions of the trees in the picture from the previous section:

[] [[]] [[][]] [[[]][]] [[[][]][[]][]]

\* \* \* \* \*

| / \ / \ /|\

\* \* \* \* \* \* \* \*

| /| |

\* \* \* \*

Note that actual input and outputs would be just the individual bracket representations. Whitespace and the ASCII art pictures are included only to make it easier to see what the trees looks like.

8.6.3 Depth

You may assume that any tree in the input is shallow enough that operating on its subtrees recursively in a sensible way will not overflow the system stack.