COMPRESSED WORDS-2

Helen has come up with a way to shorten the length of the messages send to their counterparts in the past. Helen considers only individual words, and uses the following rules to define a "shortened word":

1. a single, lower-case letter is a compressed word
2. *(e_1 e_2 ... et n)* where *t* and *n* are non-negative integers and e_i is a compressed word.

You should observe that a compressed word of one character is the same as an uncompressed word. To uncompress the compressed word *(e_1 e_2 ... e_t n)* we uncompress each e_i, concatenate those uncompressed words into a new word, and repeatedly concatenate that word *n* times. For example:

- a would be uncompressed as a,
- (a 3) would be uncompressed as aaa,
- (a (b c d 2) 3) would be uncompressed as abcdbcdabcdbcdabcdbcd.

Write a program to uncompress a compressed word.

Input

First line T i.e number of testcases(=12).Next T lines contain correctly formed **Regular Expressions.**

Output

For each case print in new line the uncompressed word **without spaces**.

Example

Input:
3

a

(a 3)

(a (b c d 2) 3)
Output:
a

aaa

abcdbcdabcdbcdabcdbcd

Constraints: 0<n<=10000, 1<=t<=26 ,

length of expressions will be less than 100 consisting of characters(a-z) and numbers.

Note:There may or may not be more than one spaces between brackets and alphabets

or number.for eg (a),(a3),(a 3) are all correct expressions.