Box Sum

You are given a string str which represents a binary tree. The string is defined using the following rules:

1. $str = TreeString$
2. $TreeString = Root[TreeString][TreeString]$
3. $Root \in$ {all english lowercase alphabets}
4. $\# = NULL$ or the end of the tree

You have to store the elements in the nodes of given tree in some labled boxes. The root of the tree will be added in the box B_0. If root of a tree is added in box B_i, its left subtree will be added in box B_{i-1} and its right subtree will be added in box B_{i+1}. And so on ...

Given the box label n and the string representation of the tree str, find the sum S of the ASCII values of the elements in the box B_n.

Input Format

One line for each test case, which contains the box label n and the string str seperated with space.

The program will stop taking input when the input str is "exit".

The number of test cases t will not exceed 500.

Constraints

$1 \leq t \leq 500$

$1 \leq |str| \leq 10000$

$-10000 \leq n \leq 10000$

Output Format

For each test case, output a new line containing string "IMPOSSIBLE" if the sum S is 0 else print one integer representing the sum S.

Sample Input 0

```
1 a[b[d[##]f[##]]c[#e[##]]]
-2 a[b[d[##]f[##]]c[#e[##]]]
0 a[b[d[##]f[##]]c[#e[##]]]
-1 l[##]
0 exit
```

Sample Output 0

```
99
100
199
IMPOSSIBLE
```

Explanation 0

For first test case, str = a[b[d[##]f[##]]c[#e[##]]] and n = 1 . After making the tree, it can be observed that B[0] will have 'af', B[1] will have 'c', B[2] will have 'e', B[-1] will have 'b' and B[-2] will have 'd'. Since n=1, we will print the sum of ascii value of characters in B[1], i.e 99 .

For second test case, str = a[b[d[##]f[##]]c[#e[##]]] and n = -2. So, we will print the sum of ascii value of characters in B[-2]. i.e 100 .

For third test case, str = a[b[d[##]f[##]]c[#e[##]]] and n = 0. So, we will print the sum of ascii value of characters in B[0]. i.e 97+102 = 199 .

For fourth test case, str = l[##] and n=-1. After making tree, only box B[0] will have 'l' and rest of the boxes will be empty. Since n=-1 and since B[-1] is empty, 'IMPOSSIBLE' will be printed without '' since sum of ascii value of characters in B[-1] is 0.


```c
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {

    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}
```

11

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code