**Even Length Palindromic Number**

You have to design a new model which maps an **even length** [palindromic number](http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=13&cad=rja&uact=8&ved=0ahUKEwjFxp7nzJvbAhVCPo8KHavhAQAQFghtMAw&url=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FPalindromic_number&usg=AOvVaw0KtkLRH2_jzn3GRtQhqxge) to some digit between 0 to 9.  
The number is mapped to a digit x on the basis of following criteria:  
1. x should appear maximum number of times in the palindromic number, that is, among all digits in the number, x should appear maximum number of times.  
2. If more than one digit appears maximum number of times, x should be the smallest digit among them.  
  
Given an integer N, you have to find the digit x for the Nth even length palindromic number.

**Note-**First 9 even length palindromic numbers are:

            11, 22, 33, 44, 55, 66, 77, 88, 99

**Input :**

First line contains **T,**number of test cases.

    Each of the next T lines contains an integer **N.**

**Output:**

For each test case, print the digit to which the Nth **even length palindromic number** is mapped.  
    Answer for each test case should come in a new line.

**Constraints:**

1≤T≤105

1≤N≤1018

**Sample Input**

3

1

2

10

**Sample Output**

1

2

0

**Explanation**

For case 1:

    1st even length palidromic number is 11 , so answer is 1 as 1 appears most number of times in the number.

For case 2:

    2nd even length palidromic number is 22 , so answer is 2 as 2 appears most number of times in the number.

For case 3:

    10th even length palindromic number is 1001, here both 0 and 1 appears same number of times but 0<1 so answer is 0.

**Note:**Your code should be able to convert the sample input into the sample output. However, this is not enough to pass the challenge, because the code will be run on multiple test cases. Therefore, your code must solve this problem statement.

Time Limit: 1.0 sec(s) for each input file

Memory Limit: 256 MB

Source Limit: 1024 KB

Marking Scheme: Marks are awarded if any testcase passes

Allowed Languages: Java, Java 8

**Minimize the value**

You are given a binary tree rooted at **1**. Initially, all the nodes of the tree have some initial values **Vi**. Wave operation is to be applied on the tree.  
  
After applying the wave operation,  
**Value of each node in the tree** = Sum of all initial values of nodes in its [subtree](https://subtree.askdefine.com/).  
  
You are required to add 1 more node with value **X**to the tree such that:  
1. The tree remains binary after adding the node to the tree.  
2. After applying the wave operation to this tree (the tree after adding node with value **X**), the sum of tree is minimum.  
  
**Sum of tree** = sum of values of all nodes in the tree.  
  
Print the minimum sum of the tree.

**Input :**

* First line of input contains 2 integers **N**and **X**, number of nodes in the tree and value of new node to add respectively.
* Second line contains **N**space separated integers denoting value of each node.
* Each of the following **N - 1** lines contains 2 integers **u**and **v**, representing edge between node **u** and node **v**.

**Output :**

* Print the minimum sum possible after adding node with value X and applying wave operation such that tree remains binary tree.

**Constraints :**

* 1≤N≤105
* 1≤Vi,X≤109
* 1≤u,v≤N, u≠v

**Sample Input**

2 1

1 1

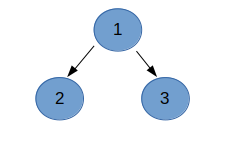
1 2

**Sample Output**

5

**Explanation**

Tree after adding new node (say it 3):



Adding node 3 to root node and applying wave operation,

Value of node 1= 1+1+1= 3 (sum of initial value of nodes 1,2 and 3 as they form the subtree of node 1)  
Value of node 2 = 1  
Value of node 3 = 1   
Sum of tree = 3+1+1 = 5 which is the minimum possible sum we can get.

**Note:**Your code should be able to convert the sample input into the sample output. However, this is not enough to pass the challenge, because the code will be run on multiple test cases. Therefore, your code must solve this problem statement.

Time Limit: 1.0 sec(s) for each input file

Memory Limit: 256 MB

Source Limit: 1024 KB

Marking Scheme: Marks are awarded if any testcase passes

Allowed Languages: Java, Java 8

**Bob and Forest**

You are given a **2D** character array denoting forest of length **N**and breadth **M**. In the matrix, **'.'**denotes barren land and **' \* '**denotes tree.  
You are given **Q**questions. In each question, you are given integer **K**and you have to determine the number of unique squares of side less than or equal to**K**which contain only trees.

**Input**

First line : *N* and *M* (*N* denoting number of rows and *M* denoting number of columns).

Each of the next ***N***lines contains string of ***M***length containing '.' and '\*' .

Next line consists of an integer *Q*, denoting number of questions.

Each of the next *Q* lines contains a single integer *K*.

**Input Constraints**

1≤N,M≤1000  
1≤Q≤105  
0≤K≤103

**Output**

For each question, print the number of unique squares of side less than or equal to**K**which contain only trees. Answer for each question should come in a new line.

**Sample Input**

4 4

\*..\*

.\*\*\*

\*\*\*\*

.\*\*\*

3

1

2

3

**Sample Output**

12

16

17

**Explanation**

In the given sample , as we can see :-

Squares of side **1**containing only trees :- 12.  
Squares of side **2**containing only trees :- 4.  
Squares of side **2**containing only trees :- 1.

Thus answer for query 1 is 12.  
Answer for query 2 i.e squares of side less than or equal to 2 are 12+4 = 16.  
Answer for query 3 i.e squares of side less than or equal to 3 are 12+4+1=17.

**Note:**Your code should be able to convert the sample input into the sample output. However, this is not enough to pass the challenge, because the code will be run on multiple test cases. Therefore, your code must solve this problem statement.

Time Limit: 1.0 sec(s) for each input file

Memory Limit: 256 MB

Source Limit: 1024 KB

Marking Scheme: Marks are awarded if any testcase passes

Allowed Languages: Java, Java 8