A. Vlad and the Best of Five

```
Input file:     standard input
Output file:    standard output
Time limit:     1 second
Memory limit:   256 megabytes
```

Vladislav has a string of length 5, whose characters are each either $\mathtt{A}$ or $\mathtt{B}$.

Which letter appears most frequently: $\mathtt{A}$ or $\mathtt{B}$?

Input

The first line of the input contains an integer t ($1 \leq t \leq 32$) — the number of test cases.

The only line of each test case contains a string of length 5 consisting of letters $\mathtt{A}$ and $\mathtt{B}$.

All t strings in a test are different (distinct).

Output

For each test case, output one letter ($\mathtt{A}$ or $\mathtt{B}$) denoting the character that appears most frequently in the string.

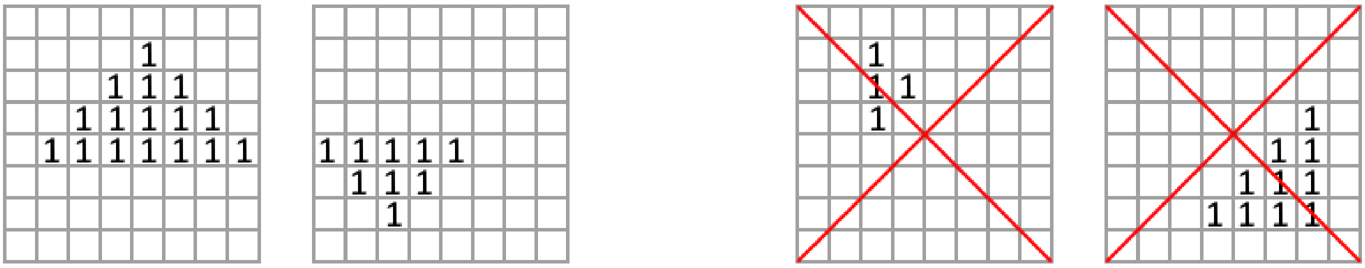
Standard Input	Standard Output
8 ABABB ABABA BBBAB AAAAA BBBBB BABAA AAAAB BAAAA	B A B A B A A A

B. Vlad and Shapes

```
Input file:     standard input
Output file:    standard output
Time limit:     1 second
Memory limit:   256 megabytes
```

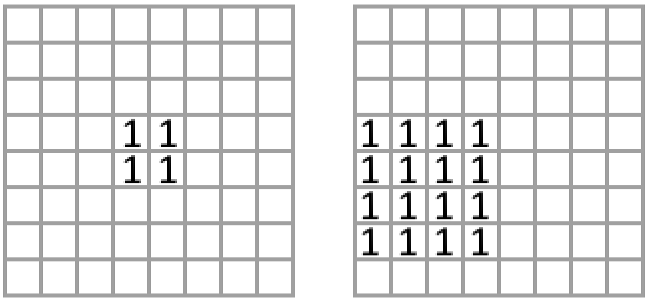
Vladislav has a binary square grid of $n \times n$ cells. A triangle or a square is drawn on the grid with symbols `1`. As he is too busy being cool, he asks you to tell him which shape is drawn on the grid.

- A *triangle* is a shape consisting of k ($k > 1$) consecutive rows, where the i-th row has $2 \cdot i - 1$ consecutive characters `1`, and the central `1`s are located in one column. An upside down triangle is also considered a valid triangle (but not rotated by 90 degrees).



Two left pictures contain examples of triangles: $k = 4$, $k = 3$. The two right pictures don't contain triangles.

- A *square* is a shape consisting of k ($k > 1$) consecutive rows, where the i-th row has k consecutive characters `1`, which are positioned at an equal distance from the left edge of the grid.



Examples of two squares: $k = 2$, $k = 4$.

For the given grid, determine the type of shape that is drawn on it.

Input

The first line contains a single integer t ($1 \leq t \leq 100$) — the number of test cases.

The first line of each test case contains a single integer n ($2 \leq n \leq 10$) — the size of the grid.

The next n lines each contain n characters `0` or `1`.

The grid contains exactly one triangle or exactly one square that contains all the `1`s in the grid. It is guaranteed that the size of the triangle or square is greater than 1 (i.e., the shape cannot consist of exactly one `1`).

Output

For each test case, output "`SQUARE`" if all the `1`s in the grid form a square, and "`TRIANGLE`" otherwise (without quotes).

Standard Input	Standard Output

6 3 000 011 011 4 0000 0000 0100 1110 2 11 11 5 00111 00010 00000 00000 00000 10 0000000000 0000000000 0000000000 0000000000 0000000000 1111111110 0111111100 0011111000 0001110000 0000100000 3 111 111 111	SQUARE TRIANGLE SQUARE TRIANGLE TRIANGLE SQUARE

,

C. Vlad and a Sum of Sum of Digits

```
Input file:     standard input
Output file:    standard output
Time limit:     0.5 seconds
Memory limit:   256 megabytes
```

Please note that the time limit for this problem is only 0.5 seconds per test.

Vladislav wrote the integers from 1 to n, inclusive, on the board. Then he replaced each integer with the sum of its digits.

What is the sum of the numbers on the board now?

For example, if $n = 12$ then initially the numbers on the board are:

$$1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12.$$

Then after the replacement, the numbers become:

$$1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3.$$

The sum of these numbers is $1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 1 + 2 + 3 = 51$. Thus, for $n = 12$ the answer is 51.

Input

The first line contains an integer t ($1 \le t \le 10^4$) — the number of test cases.

The only line of each test case contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the largest number Vladislav writes.

Output

For each test case, output a single integer — the sum of the numbers at the end of the process.

Standard Input	Standard Output
7 12 1 2 3 1434 2024 200000	51 1 3 6 18465 28170 4600002

D. Vlad and Division

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

Vladislav has n non-negative integers, and he wants to divide **all** of them into several groups so that in any group, any pair of numbers does not have matching bit values among bits from 1-st to 31-st bit (i.e., considering the 31 least significant bits of the binary representation).

For an integer k, let $k_2(i)$ denote the i-th bit in its binary representation (from right to left, indexing from 1). For example, if $k = 43$, since $43 = 101011_2$, then $43_2(1) = 1$, $43_2(2) = 1$, $43_2(3) = 0$, $43_2(4) = 1$, $43_2(5) = 0$, $43_2(6) = 1$, $43_2(7) = 0$, $43_2(8) = 0, \dots, 43_2(31) = 0$.

Formally, for any two numbers x and y in the same group, the condition $x_2(i) \neq y_2(i)$ must hold for all $1 \leq i < 32$.

What is the minimum number of groups Vlad needs to achieve his goal? Each number must fall into exactly one group.

Input

The first line contains a single integer t ($1 \leq t \leq 10^4$) — the number of test cases.

The first line of each test case contains a single integer n ($1 \leq n \leq 2 \cdot 10^5$) — the total number of integers.

The second line of each test case contains n given integers $a_1, \dots, a_n$ ($0 \leq a_j < 2^{31}$).

The sum of n over all test cases in a test does not exceed $2 \cdot 10^5$.

Output

For each test case, output a single integer — the minimum number of groups required to satisfy the condition.

Standard Input	Standard Output
9 4 1 4 3 4 2 0 2147483647 5 476319172 261956880 2136179468 1671164475 1885526767 3 1335890506 811593141 1128223362 4 688873446 627404104 1520079543 1458610201 4 61545621 2085938026 1269342732 1430258575 4 0 0 2147483647 2147483647 3 0 0 2147483647	4 1 3 2 2 3 2 2 4

8 1858058912 289424735 1858058912 2024818580 1858058912 289424735 122665067 289424735	

Note

In the first test case, any two numbers have the same last 31 bits, so we need to place each number in its own group.

In the second test case, $a_1 = 0000000000000000000000000000000_2$, $a_2 = 1111111111111111111111111111111_2$ so they can be placed in the same group because $a_1(i) \neq a_2(i)$ for each i between 1 and 31, inclusive.

,

E. Vlad and an Odd Ordering

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

Vladislav has n cards numbered $1, 2, \dots, n$. He wants to lay them down in a row as follows:

- First, he lays down all the odd-numbered cards from smallest to largest.
- Next, he lays down all cards that are twice an odd number from smallest to largest (i.e. 2 multiplied by an odd number).
- Next, he lays down all cards that are 3 times an odd number from smallest to largest (i.e. 3 multiplied by an odd number).
- Next, he lays down all cards that are 4 times an odd number from smallest to largest (i.e. 4 multiplied by an odd number).
- And so on, until all cards are laid down.

What is the k-th card he lays down in this process? Once Vladislav puts a card down, he cannot use that card again.

Input

The first line contains an integer t ($1 \le t \le 5 \cdot 10^4$) — the number of test cases.

The only line of each test case contains two integers n and k ($1 \le k \le n \le 10^9$) — the number of cards Vlad has, and the position of the card you need to output.

Output

For each test case, output a single integer — the k-th card Vladislav lays down.

Standard Input	Standard Output
11 7 1 7 2 7 3 7 4 7 5 7 6 7 7 1 1 34 14 84 19 1000000000 1000000000	1 3 5 7 2 6 4 1 27 37 536870912

Note

In the first seven test cases, $n = 7$. Vladislav lays down the cards as follows:

- First — all the odd-numbered cards in the order 1, 3, 5, 7.
- Next — all cards that are twice an odd number in the order 2, 6.

- Next, there are no remaining cards that are 3 times an odd number. (Vladislav has only one of each card.)
- Next — all cards that are 4 times an odd number, and there is only one such card: 4.
- There are no more cards left, so Vladislav stops.

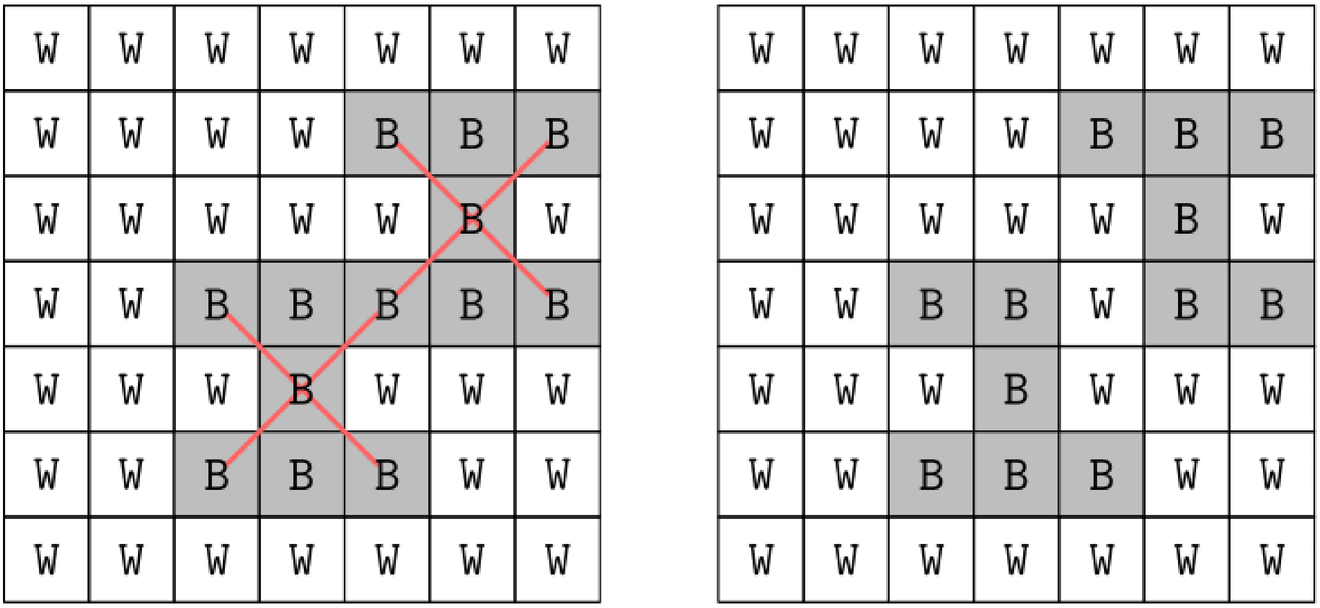
Thus the order of cards is 1, 3, 5, 7, 2, 6, 4.

F. Vlad and Avoiding X

```
Input file:     standard input
Output file:    standard output
Time limit:     4 seconds
Memory limit:   256 megabytes
```

Vladislav has a grid of size 7×7, where each cell is colored black or white. In one operation, he can choose any cell and change its color (black $\leftrightarrow$ white).

Find the minimum number of operations required to ensure that there are no black cells with four diagonal neighbors also being black.



The left image shows that initially there are two black cells violating the condition. By flipping one cell, the grid will work.

Input

The first line of input contains a single integer t ($1 \leq t \leq 200$) — the number of test cases. Then follows the description of the test cases.

Each test case consists of 7 lines, each containing 7 characters. Each of these characters is either W or B, denoting a white or black cell, respectively.

Output

For each test case, output a single integer — the minimum number of operations required to ensure that there are no black cells with all four diagonal neighbors also being black.

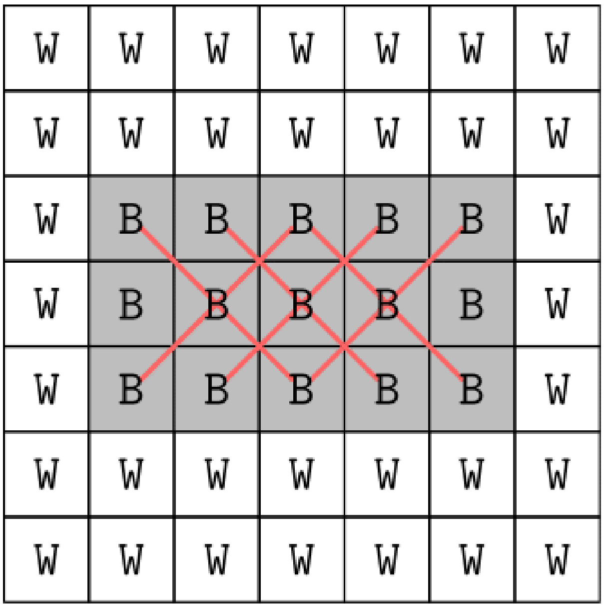
Standard Input	Standard Output
4 WWWWWWW WWWWBBB WWWWWBW WWBBBBB WWWBWWW WWBBBWW WWWWWWW WWWWWWW	1 2 0 5

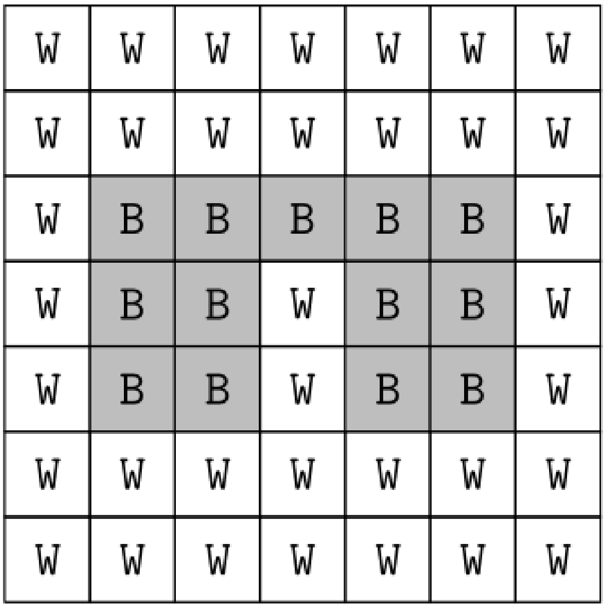
WWWWWWW WBBBBBW WBBBBBW WBBBBBW WWWWWWW WWWWWWW WWWWWWW WWWWWWW WWWWWWW WWWWWWW WWWWWWW WWWWWWW WWWWWWW WBBBBBW BBBBBBB BBBBBBB WWWWWWW BBBBBBB BBBBBBB BBBBBBB	

Note

The first test case is illustrated in the statement.

The second test case is illustrated below:





In the third test case, the grid already satisfies the condition.

G. Vlad and Trouble at MIT

```
Input file:     standard input
Output file:    standard output
Time limit:     1 second
Memory limit:   256 megabytes
```

Vladislav has a son who really wanted to go to MIT. The college dormitory at MIT (Moldova Institute of Technology) can be represented as a tree with n vertices, each vertex being a room with exactly one student. A *tree* is a connected undirected graph with n vertices and $n-1$ edges.

Tonight, there are three types of students:

- students who want to party and play music (marked with $\mathtt{P}$),
- students who wish to sleep and enjoy silence (marked with $\mathtt{S}$), and
- students who don't care (marked with $\mathtt{C}$).

Initially, all the edges are *thin* walls which allow music to pass through, so when a partying student puts music on, it will be heard in every room. However, we can place some *thick* walls on any edges — thick walls don't allow music to pass through them.

The university wants to install some thick walls so that every partying student can play music, and no sleepy student can hear it.

Because the university lost a lot of money in a naming rights lawsuit, they ask you to find the minimum number of thick walls they will need to use.

Input

The first line contains a single integer t ($1 \le t \le 1000$) — the number of test cases.

The first line of each test case contains an integer n ($2 \le n \le 10^5$) — the number of vertices in the tree.

The second line of each test case contains $n-1$ integers $a_2, \ldots, a_n$ ($1 \le a_i < i$) — it means there is an edge between i and a_i in the tree.

The third line of each test case contains a string s of length n consisting of characters $\mathtt{P}$, $\mathtt{S}$, and $\mathtt{C}$, denoting that student i is of type s_i.

The sum of n over all test cases does not exceed 10^5.

Output

For each test case, output a single integer — the minimum number of thick walls needed.

Standard Input	Standard Output
3 3 1 1 CSP 4 1 2 2 PCSS 4 1 2 2	1 1 2

PPSS	

Note

In the first case, we can install one thick wall between rooms 1 and 2, as shown below. We cannot install 0 walls, since then the music from room 3 will reach room 2 where a student wants to sleep, so the answer is 1. There are other valid solutions.