A: Sum of Two Cards

Time Limit: 1 second

Description

This is a simplified version of the game blackjack! Players are given two random cards from the deck. The players add the values of the cards. The player with the highest sum, wins!

In general, given two cards, a and b, return their sum.

Remember, a deck of cards contains numbers from 2 to 10 and letters A (ace), J (jack), Q (queen), and K (king). The letter A has a value of 1, J is 11, Q is 12, and K is 13.

Input Format

The input starts with a positive integer d that tells you the number of players in the game. Succeeding d lines contain two cards a and b.

Output Format

Return the sum of the values of the two cards for all players.

Constraints

$0 \leq d \leq 26$

$a, b = \{2..10, A, J, Q, K\}$

Sample Input

```
5
2 J
K Q
10 5
Q Q
A 9
```

Sample Output

```
13
25
15
24
10
```

B: Variations of Sign

Time limit: 1 second

Description

If the terms of a polynomial $f(x)$ with real coefficients are written in descending powers of x, a variation in sign is said to occur whenever two consecutive terms differ in sign. The missing powers of x are disregarded in counting the number of variations of sign.

Examples:

$f(x) = x^3 + x^2 - 10x + 8$ has 2 variations of sign.
$f(x) = x^3 - 2x^2 + 3x - 12$ has 3 variations of sign.

Constraints

$2 \leq n \leq 10\,000$, where n is the order of the polynomial $f(x)$.

Input Format

The input starts with a positive integer t representing the total number of test cases. This is followed by t lines each containing an integer n, representing the order (i.e. highest degree) of polynomial $f(x)$ followed by $n + 1$ integers representing the numerical coefficients of the terms in $f(x)$ starting from the highest degree down to the lowest.

Output Format

For each case, the output must be of the form:

CASE k: v

where k is the case number (starting from 1) and v is the total number of variations in sign of the corresponding polynomial.

Sample Input

```
3
3 1 1 -10 8
3 1 -2 3 -12
4 1 -3 3 0 5
```

Sample Output

```
CASE 1: 2
CASE 2: 3
CASE 3: 2
```

C: Hey Irie Live!

Time Limit: 1 second

Description

Your favorite band **Hey Irie** will be having their first intimate concert. To expedite the ticketing system, you decided to create a program to assign the audience seats with their corresponding prices. The layout of the venue looks like this:

```
1 A B C D E F G H
2 A B C D E F G H
3 A B C D E F G H
4 A B C D E F G H
5 A B C D E F G H
6 A B C D E F G H
7 A B C D E F G H
8 A B C D E F G H
9 A B C D E F G H
```

The program should display the seat pattern with an **X** marking the seats already assigned. For example, after seats 4D, 7A, and 9E are taken, the display should look like this:

```
1 A B C D E F G H
2 A B C D E F G H
3 A B C D E F G H
4 A B C X E F G H
5 A B C D E F G H
6 A B C D E F G H
7 X B C D E F G H
8 A B C D E F G H
9 A B C D X F G H
```

Input Format

Input begins with an integer N indicating the number of seat requests. Succeeding are N lines of input in the format nc, where n indicates the row number and c indicates the column letter of the seat being requested.

Output Format

If the requested seat is available, display the seat view as shown above. Mark the reservation with an uppercase '**X**'. If the requested seat is not available, print "**Sorry, seat is taken!**" without the quotation marks. Separate output for each passenger request with an empty line. Assume all seats are available at the start of the program. Then, display the total payable cost of the tickets after the reservations made. The pricing is as follows:

1	A	B	C	D	E	F	G	H	
2	A	B	C	D	E	F	G	H	
3	A	B	C	D	E	F	G	H	ticket prices:
4	A	B	C	D	E	F	G	H	3,000
5	A	B	C	D	E	F	G	H	2,000
6	A	B	C	D	E	F	G	H	1,500
7	A	B	C	D	E	F	G	H	
8	A	B	C	D	E	F	G	H	
9	A	B	C	D	E	F	G	H	

Constraints
$1 \leq N \leq 100$
$1 \leq n \leq 7$
$c = \{A, B, C, D, E, F, G, H\}$

Sample Input
4
3C
5G
7A
5G

Sample Output
1 A B C D E F G H
2 A B C D E F G H
3 A B X D E F G H
4 A B C D E F G H
5 A B C D E F G H
6 A B C D E F G H
7 A B C D E F G H
8 A B C D E F G H
9 A B C D E F G H

1 A B C D E F G H
2 A B C D E F G H
3 A B X D E F G H
4 A B C D E F G H
5 A B C D E F X H
6 A B C D E F G H
7 A B C D E F G H
8 A B C D E F G H
9 A B C D E F G H

```
1 A B C D E F G H
2 A B C D E F G H
3 A B X D E F G H
4 A B C D E F G H
5 A B C D E F X H
6 A B C D E F G H
7 X B C D E F G H
8 A B C D E F G H
9 A B C D E F G H

Sorry, seat is taken!

Total: 6500
```

D: Emotions are for People

Time Limit: 1 second

Description

In the corporate world, there is no room for emotions. This is not a sustainable state of mind for most people and that is why many in the management field experience stress, some overly so.

Take a manager who is asked to fire the most unproductive member of his team. This person must consider the life of the one he is set to fire: the family, responsibilities and the dreams. But in the in this world it is either him or his employee. His conscience can't take it. He feels like he will break, so he builds a program instead. Let the machine handle this, perhaps he can have a good night sleep for once.

This program reads employee records and recommends who gets fired. Dreams be damned. It is a simple mathematical truth: the highest ranking, oldest, highest paid and who receives the most benefits gets fired first. The manager thinks this will save the most money for the company. Maybe it will make his boss happier. Maybe.

He gave this program to his boss.

He doesn't know.

He went home relieved.

He still doesn't know.

Later that night, he could not go to sleep.

Then he knew.

He was going to get fired tomorrow.

Input Format

The first line contains a positive integer n that represents the number of employees to process. Succeeding lines consist of 8 numbers separated by spaces. These 8 numbers represent employee information, namely: ID number a, rank b, age c, years of service d, company salary e, revenue generated f, commissions received from customer g, and monetary benefits from the company received by a single employee h.

Output Format

The program outputs the string "Employee " followed by the fired employee's ID number, then the string ", you are fired!".

Constraints

$0 \leq n$
$0 \leq a, b, c, d, e, f, g, h \leq 1\ 000\ 000$

Sample Input

3
201103041 1 29 8 1000 12000 2000 6000
200010231 2 33 12 1200 2000 500 3000
199819246 1 40 15 1800 5000 300 3000

Sample Output

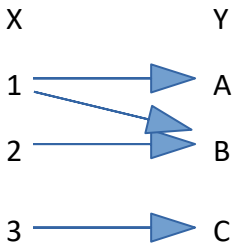
Employee 200010231, you are fired!

E: Functions and Relations

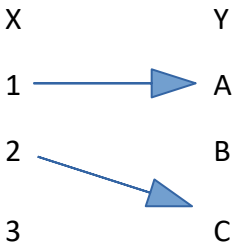
Time Limit: 1 second

Description

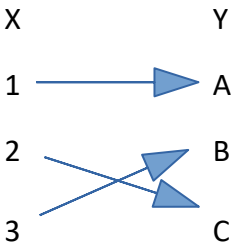
All functions are relations. But not all relations are functions. Let's say you have set $X = \{1,2,3\}$ and you have set $Y = \{A, B, C\}$. A function from X to Y is a relation from X to Y such that for each element in X, there exists exactly one element matched for it in Y. Study the following diagrams:



Not a function. Some X has more than one match in Y.



Not a function. Some X do not have a match in Y.



A function. Each X has exactly one match in Y.

Input Format

Assume you have set $X = \{0,1,2,3,4,5,6,7,8,9\}$ and set Y. The first line contains the number of data $d > 0$ to process. Succeeding n lines consist of exactly 10 integer values belonging to set X which has a match in set Y. If a value repeats, it means it has another match in Y, and thus, the relation from X to Y is not a function. Likewise, if a value in X is missing, then it doesn't have a match and thus, the relation from X to Y is not a function. If all values in X are represented without repetition, then it means each has exactly one match in Y and hence, the relation from X to Y is a function.

Output Format

For each line, display "FUNCTION" if the values make the relation from X to Y a function. Otherwise, display "Relation".

Sample Input

```
5
9 3 5 4 1 0 6 8 2 7
4 8 1 0 8 2 9 1 7 8
4 1 0 2 7 5 8 6 3 9
2 6 9 4 8 4 7 1 1 8
0 7 8 0 7 3 6 4 5 3
```

Sample Output

```
FUNCTION
Relation
FUNCTION
Relation
Relation
```

F: LLPS

Time Limit: 3 seconds

Description

A sequence of characters which reads the same backward as forward is called a *palindrome*. The word **racecar**, for example, is read the same when spelled backwards. A *substring*, on the other hand, is a sequence of contiguous characters contained within a string. For example, the words **a**, **ace**, **car**, **race**, and **racecar** are all substrings of the word racecar.

Any non-empty string, which may or may not be palindrome, surely has at least one palindrome substring in it. Take for example the word **supercalifragilisticexpialidocious**, although obviously not a palindrome, it consists of thirty-four one-character palindrome substrings (since the word contains thirty four (34) characters and a single character is evidently a palindrome) as well as one three-character palindrome substring, **ili**. Since the latter is the longest palindrome substring, we say that the length of the longest palindrome substring in the word supercalifragilisticexpialidocious is three (3).

Given a string $s = s_0 s_1 \dots s_{n-2} s_{0n-1}$ of length n, where s_i represent characters, find the length of the longest palindrome substring contained within it.

Constraints

s contains only lowercase or uppercase alphabet letters.
$0 \leq n \leq 100$, where n is the length of strings.

Input Format

The input starts with a positive integer t representing the total number of test cases. This is followed by t lines each containing a string s.

Output Format

For each case, the output must be of the form:

$$CASE\ k: L$$

where k is the case number (starting from 1) and L is the length of the corresponding string s in the input.

Sample Input	**Sample Output**
3	CASE 1: 1
Ace	CASE 2: 3
Supercalifragilisticexpialidocious	CASE 3: 7
racecar	

G: Staircase

Time Limit: 1 second

Description

What took you so long? Solve this quick and go up the leaderboard. 😊

There is a staircase with N steps. With the limitation of the length of your legs, you can only climb 1 step or 2 steps at a time. Your goal is to know the unique number of ways you can climb the staircase. Note: The order of the steps matter.

For example, if N is 3, then there are 3 unique ways: $[1,2], [1,1,1], [2,1]$. If N is 4, then there are 5 ways.

Input Format

First line contains the number of test cases t. The succeeding lines contain t number of steps N.

Output Format

Display the number of unique ways you can climb the staircase.

Constraints

$0 < t, n < 100$

Sample Input	**Sample Output**
2	3
3	34
8	