Problem A
Average Character
Time Limit: 1 Second(s)

Have you ever wondered what the average ASCII character of any given string is? No? Never? Really? Well, is it a character in the string or something else?

Would you do this calculation by hand with an ASCII table? Probably not! All modern programming languages include functions for converting an ASCII character to an integer, and to convert an integer to an ASCII character. Of course, these functions often also handle unicode characters as well, but that is not part of this problem.

Given a string of ASCII characters, compute the average character. If the average character lies between two integer ASCII values, return the smaller one.

Input

The single line of input contains a single string s ($1 \leq |s| \leq 100$), which consists of ASCII text. All of the characters of s will be printable ASCII, between ASCII 32 (space: `' '`) and ASCII 126 (tilde: `'~'`). It will **NOT** contain any control characters such as carriage returns, line feeds, tabs, etc. It is **NOT** guaranteed to begin, end, or even contain a non-space character.

Output

Output a single ASCII character, which is the average of all of the ASCII characters in s.

Sample Input 1	Sample Output 1
`ABCDE`	`C`

Sample Input 2	Sample Output 2
`AbCdE`	`O`

Sample Input 3	Sample Output 3
`aBcDe`	`V`

This page is intentionally left blank.

Problem B
Black and White
Time Limit: 1 Second(s)

Black and White is a Chinese children's game played in rounds. During each round, the children who are playing all put their hands in either face-up ("White") or face-down ("Black"). If all the children but one make the same choice, then the "odd one out" sits out for the rest of the game. Play continues until there are only two children left.

Each child independently chooses whether to put their hand face-up with their own fixed probability. What is the expected number of rounds that such a game will last?

Input

The first line contains a single integer n ($2 \le n \le 20$), which is the number of children.

Each of the next n lines contains a single real number p ($0.1 \le p \le 0.9$). These are the probabilities for each child that they will put their hand in face-up. The probabilities will have at most three digits after the decimal point.

Output

Output a single real number, which is the expected number of rounds. The result must be accurate to within an absolute or relative error of 10^{-6}.

Sample Input 1	Sample Output 1
3 0.5 0.5 0.5	1.3333333

Sample Input 2	Sample Output 2
3 0.3 0.3 0.3	1.5873015

Sample Input 3	Sample Output 3
5 0.1 0.3 0.5 0.7 0.9	7.4752846

Problem C
Problem Set Construction
Time Limit: 1 Second(s)

You are a judge, constructing a problem set for a contest. You have a pool of candidate problems. For each problem, you've found the probability that a team is able to solve the problem, and the time it will take them to implement the solution if they are able to solve it. All implementation times are distinct.

You know the strategy that all teams will take when confronted with a problem set. First, they will determine the set of problems they can solve (assume they can do this instantly at the beginning of the contest). Then, they will solve as many of those problems as they can under the time limit. If there are many subsets of problems they can solve under the time limit, they will first break ties by the number of problems they can solve, next they will break ties by minimizing the total time it will take to solve all of those problems.

Define the *Difficulty* of a problem to be the probability that a team will solve the problem if it is included in a problem set of size k along with $k-1$ other problems chosen uniformly at random from the pool. Find the *Difficulties* of all the problems.

Input

The first line of input contains three integers n, k ($1 \le k \le n \le 50$) and t ($1 \le t \le 2500$), where n is the number of problems in the pool, k is the number of problems to be chosen for the set, and t is the time limit of the contest.

Each of the next n lines contains a real number p ($0.0 \le p \le 1.0$) and an integer s ($1 \le s \le t$) describing a problem, where p is the probability that a team is able to solve it, and s is the time to solve. The probabilities will have at most four decimal digits. All times to solve will be distinct.

Output

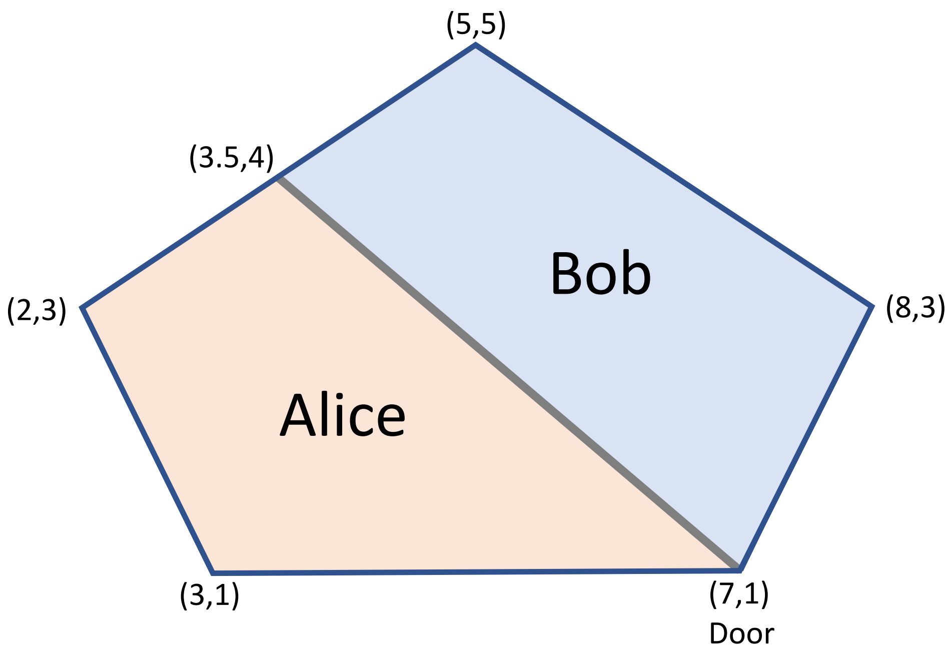
Output n lines, each containing a real number which is the *Difficulty* of the given problem in the order of the input. Each value must be accurate to within an absolute or relative error of 10^{-6}.

Sample Input 1	Sample Output 1
3 1 100 0.3432 99 0.1231 100 0.5878 1	0.343200 0.123100 0.587800

Sample Input 2	Sample Output 2
3 2 100 0.3432 99 0.1231 100 0.5878 2	0.242334 0.065797 0.587800

Problem D
Dorm Room Divide
Time Limit: 1 Second(s)

Bob and Alice are roommates at the International College of Polygonal Chambers (ICPC). To avoid conflict, they've agreed to divide their dorm room in half—as closely as possible. However, the room is shaped so irregularly that they need your help!



Each dorm room is a convex polygon, with a single entrance. You need to figure out how to divide this room in half (by area) using a single straight line starting at the door, and terminating on a wall or corner of the room.

Input

The first line of input contains a single integer n ($3 \leq n \leq 2 \cdot 10^5$), which is the number of vertices describing the convex polygon.

Each of the next n lines contains two space-separated integers x and y ($-10^7 \leq x, y \leq 10^7$). These are the coordinates of the vertices of the convex polygon, in counterclockwise order. All points will be distinct.

The door is considered to be a single point located at the first vertex given in the input.

Output

Output two space-separated real numbers, which are the x and y coordinates of the other endpoint of the dividing line, such that the area of the room is divided in half. Each coordinate value must be accurate to within an absolute or relative error of 10^{-6}. Output x first, then y.

Note that Sample 1 corresponds to the example in the problem description.

Sample Input 1

```
5
7 1
8 3
5 5
2 3
3 1
```

Sample Output 1

```
3.5 4
```

Sample Input 2

```
3
2 2
10 3
6 8
```

Sample Output 2

```
8 5.5
```

Problem E
Overdraft
Time Limit: 1 Second(s)

Banks often charge overdraft fees if you attempt to withdraw more money from your account than is available in your current balance. Given a sequence of deposits and withdrawals (and assuming each deposit and withdrawal is immediately reflected in your balance), determine the minimum (non-negative) starting balance you need to ensure that you will not be charged any overdraft fees over the course of the sequence.

Input

The first line of input contains a single integer n ($1 \leq n \leq 1{,}000$), which is the number of transactions.

Each of the next n lines contains a single integer t ($-10^6 \leq t \leq 10^6$, $t \neq 0$). These are the transactions, in the order that they occur. A positive number represents a deposit, a negative number represents a withdrawal. No two transactions occur simultaneously.

Output

Output a single non-negative integer, which is the minimum non-negative balance you must start with in your account in order to avoid any overdraft fees.

Sample Input 1	Sample Output 1
3 3 -5 3	2

This page is intentionally left blank.

Problem F
Fail Them All!
Time Limit: 3 Second(s)

You are an instructor for an algorithms course, and your students have been saying mean things about you on social media. Those jerks! Being a vengeful and dishonest instructor, you are going to make them pay.

You have given your students a True/False exam. For each question, each student is allowed to either answer the question or leave the question blank. Each student has answered at least two questions. You want to make sure that every student fails the test, so you are going to alter the answer key so that no student gets more than one answer correct.

Is there an answer key such that every person has at most one submitted answer that is correct? If so, compute the lexicographically minimal such answer key.

Input

The first line of input contains two integers n ($1 \le n \le 100$) and k ($2 \le k \le 100$), where n is the number of students in the class, and k is the number of questions on the test.

Each of the next n lines contains a string s ($|s| = k$, $s \in \{\mathtt{T}, \mathtt{F}, \mathtt{X}\}^*$), which are the answers to the questions, in order, for each student, where `T` means True, `F` means False, and `X` means the student didn't answer the question. Every student's answers will have at least two which are not `X`.

Output

If such an answer key can be constructed, output a string of length k consisting of only the characters `T` and `F`, which is the answer key. If more than one such key is possible, output the one which comes first alphabetically (`F` < `T`). If no such key exists, instead output -1.

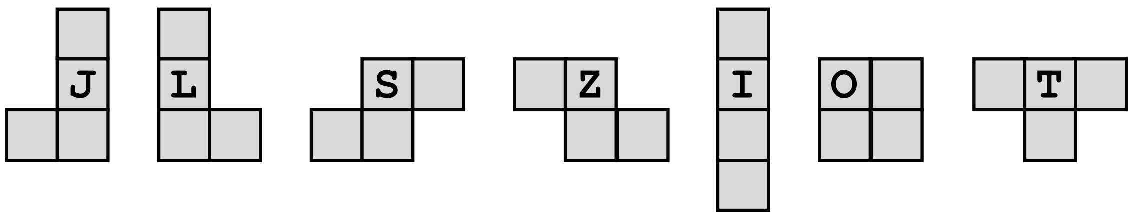
Sample Input 1	Sample Output 1
3 3 FFX XFF FXF	FTT

Sample Input 2	Sample Output 2
3 3 FTX XFT TXF	FFF

Sample Input 3	Sample Output 3
4 3 TTX XTT TXT FFF	-1

Problem G
Tetris Generation
Time Limit: 1 Second(s)

The classic game Tetris involves arranging falling tetrominoes on a board. There are seven different tetrominoes, each named after a letter that resembles their shape: `J`, `L`, `S`, `Z`, `I`, `O`, and `T`.



In the original Tetris, the player would receive one tetromino at a time, and each tetromino would be chosen from among the seven possibilities independently and uniformly at random. This meant that any sequence of tetrominoes could appear in a game, such as numerous `I` tetrominoes in a row. Modern versions of Tetris remove these streaks by generating tetrominoes in groups of seven: The first seven tetrominoes in a game will be one of each of the seven different tetrominoes in a random order. The next seven tetrominoes will also be one of each of the seven different tetrominoes in a random order (possibly but not necessarily different from the ordering of the first seven). Same goes for the next seven, and so on and so forth. With this generator, it is still possible to get two of the same tetromino in a row (for example, the seventh and eighth tetrominoes in the game can be the same as each other), but it is not possible to get three of the same type in a row.

Given a sequence of tetrominoes, determine whether it is possible for a modern Tetris generator to produce that sequence at some point in a game.

Input

The first line of input contains an integer t ($1 \le t \le 10^5$), which is the number of test cases.

Each of the next t lines contains a single string s ($1 \le |s| \le 1{,}000$, $s \in \{\mathtt{J}, \mathtt{L}, \mathtt{S}, \mathtt{Z}, \mathtt{I}, \mathtt{O}, \mathtt{T}\}^*$). This string represents a sequence of tetrominoes, and is a single test case.

The sum of the lengths of all input test cases will not exceed 10^5.

Output

For each test case, output a single line with a single integer, which is `1` if the sequence can be generated by a modern Tetris generator, and `0` otherwise.

Sample Input 1	Sample Output 1
2 JJTO JJTT	1 0

Problem H
Tree Hopping
Time Limit: 1 Second(s)

You are given a tree and a permutation of its vertices. It can be proven that for any tree and any pair of source/destination nodes, there is some permutation of the nodes where the first node is the source, the last node is the destination, and the distance between adjacent nodes in the permutation is less than or equal to three.

Your job will be to write a verifier for this property. Given such a permutation and the tree, validate whether the distance between adjacent nodes in the permutation is less than or equal to three.

Input

The first line of input contains an integer t ($1 \leq t \leq 50{,}000$), which is the number of test cases.

In each test case, the first line of input contains an integer n ($2 \leq n \leq 100{,}000$), which is the number of nodes in the tree. The nodes are numbered from 1 to n.

Each of the next $n-1$ lines contains a pair of integers a and b ($1 \leq a < b \leq n$), representing an edge in the tree between nodes a and b.

Each of the next n lines contains an integer p ($1 \leq p \leq n$, all values distinct). This is the permutation of the nodes.

The sum of the values of n over all test cases will not exceed $100{,}000$.

Output

For each test case, output a single line with a single integer, which is 1 if the given permutation satisfies the constraint that every pair of adjacent nodes in the permutation has distance less than or equal to three in the tree. Output 0 if the given permutation does not satisfy this constraint.

Sample Input 1	Sample Output 1
2 5 1 2 2 3 3 4 4 5 1 3 2 5 4 5 1 2 2 3 3 4 4 5 1 5 2 3 4	1 0

Problem I
XOR Island
Time Limit: 5 Second(s)

On an island populated entirely with perfect logicians, each islander is wearing a hat that displays a positive integer. Each islander can see all other islanders' hats, but they cannot see their own hat. An islander has no information about the number on their own hat, other than the fact that it is a positive integer.

One day, a mysterious message appears in the sky and says "There exist three distinct islanders such that the XOR of the integers on two of their hats is the integer on the hat of the third". After this message appears, the islanders schedule meetings for several days in a row, one meeting per day. At each meeting any islander who knows for sure that they are part of some triple that satisfies the message will raise their hand.

Assuming that no islanders lie or make mistakes, and each will raise their hand as soon as it is possible for them to know that they are part of a triple, how many days will it take for at least one islander to raise their hand at a meeting?

Input

The first line of input contains a single integer n ($3 \leq n \leq 25$), the number of islanders.

Each of the next n lines contains a single integer a ($1 \leq a < 2^{25}$), which are the positive integers on the islanders' hats.

It is guaranteed that the input is chosen such that the message in the sky is true; there is at least one triple among the islanders' hats such that one hat's integer is equal to the XOR of the integers on the other two hats.

Output

Output a single integer, which is the number of daily meetings it will take for some islander to figure out that their hat is part of some XOR triple. It can be proven that at least one person will eventually raise their hand.

Sample Input 1

```
3
1
2
3
```

Sample Output 1

```
1
```

Sample Input 2

```
11
9
1
14
2
11
7
6
7
6
5
3
```

Sample Output 2

```
3
```

Problem J
Who Goes There?
Time Limit: 1 Second(s)

What happens when more teams want to go to an ICPC regional site than the site has capacity for? Who goes there?

One possible policy is the following: Every school is allowed to register as many teams as they wish. Accept every school's first team, then accept every school's second team (for schools with more than one team), then third, and so on, until all teams are accepted, or there isn't enough capacity for the next wave. Then, if there are extra spots available, the spots are given to schools, one by one, in the order that the schools registered.

Given the capacity of a site, the number of teams registered by each school and the order that they registered, determine how many teams from each school are accepted.

Input

The first line of input contains two integers n ($1 \le n \le 100$) and m ($1 \le m \le 100$), where n is the capacity of the site and m is the number of schools that wish to compete there.

Each of the next m lines contains an integer t ($1 \le t \le 100$), which is the number of teams that a school has registered. The schools are listed in the order that they registered.

Output

Output m lines, one for each school. Each line must contain a single integer indicating the number of teams accepted from that school. Output them in the same order as they appear in the input.

Sample Input 1	Sample Output 1
20 5 7 5 1 6 12	5 5 1 5 4

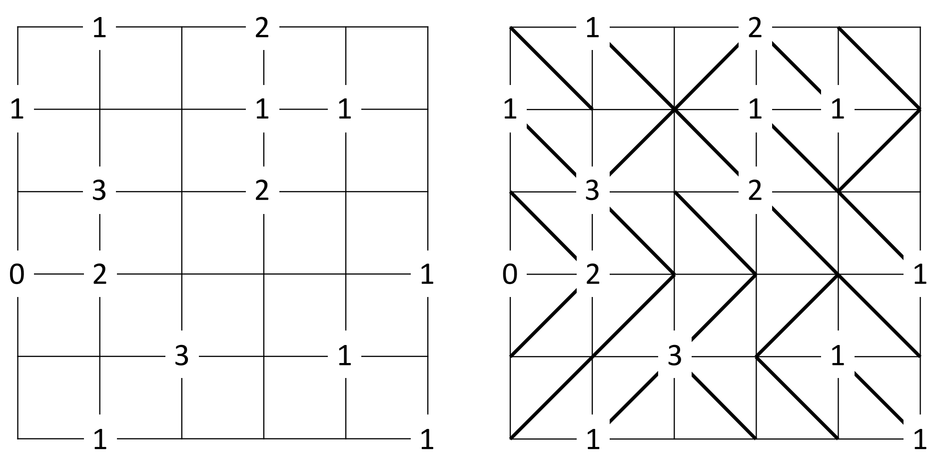
This page is intentionally left blank.

Problem K
Diagonals
Time Limit: 10 Second(s)

Diagonals is a pencil puzzle which is played on a square grid. The player must draw a diagonal line corner to corner in every cell in the grid, either top left to bottom right, or bottom left to top right. There are two constraints:

- Some intersections of gridlines have a number from 0 to 4 inclusive on them, which is the exact number of diagonals that must touch that point.
- No set of diagonals may form a loop of any size or shape.

The following is a 5×5 example, with its unique solution:



Given the numbers at the intersections of a grid, solve the puzzle.

Input

The first line of input contains an integer n ($1 \leq n \leq 8$), which is the size of the grid.

Each of the next $n+1$ lines contains a string s ($|s| = n+1, s \in \{0, 1, 2, 3, 4, +\}^*$). These are the intersections of the grid, with `+` indicating that there is no number at that intersection.

The input data will be such that the puzzle has exactly one solution.

Output

Output exactly n lines, each with exactly n characters, representing the solution to the puzzle. Each character must be either '/' or '\'.

Note that Sample 1 corresponds to the example in the problem description.

Sample Input 1

```
5
+1+2++
1++11+
+3+2++
02+++1
++3+1+
+1+++1
```

Sample Output 1

```
\\/\\
\/\\/
\\\\\
////\
//\\\
```

Sample Input 2

```
3
++++
+1+1
+31+
+0+0
```

Sample Output 2

```
/\/
///
/\/
```

Sample Input 3

```
4
+++++
+3++2
++3++
+3+3+
++2+0
```

Sample Output 3

```
\//\
\\//
\\\/
/\//
```