Problem J1: Winning Score

Problem Description
You record all of the scoring activity at a basketball game. Points are scored by a 3-point shot, a 2-point field goal, or a 1-point free throw.

You know the number of each of these types of scoring for the two teams: the Apples and the Bananas. Your job is to determine which team won, or if the game ended in a tie.

Input Specification
The first three lines of input describe the scoring of the Apples, and the next three lines of input describe the scoring of the Bananas. For each team, the first line contains the number of successful 3-point shots, the second line contains the number of successful 2-point field goals, and the third line contains the number of successful 1-point free throws. Each number will be an integer between 0 and 100, inclusive.

Output Specification
The output will be a single character. If the Apples scored more points than the Bananas, output `A`. If the Bananas scored more points than the Apples, output `B`. Otherwise, output `T`, to indicate a tie.

Sample Input 1
```
10
3
7
8
9
6
```

Output for Sample Input 1
```
B
```

Explanation of Output for Sample Input 1
The Apples scored $10 \cdot 3 + 3 \cdot 2 + 7 \cdot 1 = 43$ points and the Bananas scored $8 \cdot 3 + 9 \cdot 2 + 6 \cdot 1 = 48$ points, and thus the Bananas won.

Input for Sample Input 2
```
7
3
0
6
4
1
```

La version française figure à la suite de la version anglaise.

Output for Sample Input 2

```
T
```

Explanation of Output for Sample Input 2

The Apples scored $7 \cdot 3 + 3 \cdot 2 + 0 \cdot 1 = 27$ points and the Bananas scored $6 \cdot 3 + 4 \cdot 2 + 1 \cdot 1 = 27$ points, and thus it was a tie game.

La version française figure à la suite de la version anglaise.

Problem J2: Time to Decompress

Problem Description

You and your friend have come up with a way to send messages back and forth.

Your friend can encode a message to you by writing down a positive integer N and a symbol. You can decode that message by writing out that symbol N times in a row on one line.

Given a message that your friend has encoded, decode it.

Input Specification

The first line of input contains L, the number of lines in the message.

The next L lines each contain one positive integer less than 80, followed by one space, followed by a (non-space) character.

Output Specification

The output should be L lines long. Each line should contain the decoding of the corresponding line of the input. Specifically, if line $i + 1$ of the input contained `N x`, then line i of the output should contain just the character `x` printed `N` times.

Sample Input

```
4
9 +
3 -
12 A
2 X
```

Output for Sample Input

```
+++++++++
---
AAAAAAAAAAAA
XX
```

La version française figure à la suite de la version anglaise.

Problem J3: Cold Compress

Problem Description

Your new cellphone plan charges you for every character you send from your phone. Since you tend to send sequences of symbols in your messages, you have come up with the following compression technique: for each symbol, write down the number of times it appears consecutively, followed by the symbol itself. This compression technique is called *run-length encoding*.

More formally, a block is a substring of identical symbols that is as long as possible. A block will be represented in compressed form as the length of the block followed by the symbol in that block. The encoding of a string is the representation of each block in the string in the order in which they appear in the string.

Given a sequence of characters, write a program to encode them in this format.

Input Specification

The first line of input contains the number N, which is the number of lines that follow. The next N lines will contain at least one and at most 80 characters, none of which are spaces.

Output Specification

Output will be N lines. Line i of the output will be the encoding of the line $i + 1$ of the input. The encoding of a line will be a sequence of pairs, separated by a space, where each pair is an integer (representing the number of times the character appears consecutively) followed by a space, followed by the character.

Sample Input

```
4
+++===!!!!
777777......TTTTTTTTTTTT
(AABBC)
3.1415555
```

Output for Sample Input

```
3 + 3 = 4 !
6 7 6 . 12 T
1 ( 2 A 2 B 1 C 1 )
1 3 1 . 1 1 1 4 1 1 4 5
```

Explanation of Output for Sample Input

To see how the first message (on the second line of input) is encoded, notice that there are 3 + symbols, followed by 3 = symbols, followed by 4 `!` symbols.

La version française figure à la suite de la version anglaise.

Problem J4/S1: Flipper

Problem Description
You are trying to pass the time while at the optometrist. You notice there is a grid of four numbers:

1	2
3	4

You see lots of mirrors and lenses at the optometrist, and wonder how flipping the grid horizontally or vertically would change the grid.

Specifically, a "horizontal" flip (across the horizontal centre line) would take the original grid of four numbers and result in:

3	4
1	2

A "vertical" flip (across the vertical centre line) would take the original grid of four numbers and result in:

2	1
4	3

Your task is to determine the final orientation of the numbers in the grid after a sequence of horizontal and vertical flips.

Input Specification
The input consists of one line, composed of a sequence of at least one and at most 1 000 000 characters. Each character is either `H`, representing a horizontal flip, or `V`, representing a vertical flip.

For 8 of the 15 available marks, there will be at most 1 000 characters in the input.

Output Specification
Output the final orientation of the four numbers. Specifically, each of the two lines of output will contain two integers, separated by one space.

Sample Input 1
```
HV
```

Output for Sample Input 1
```
4 3
2 1
```

La version française figure à la suite de la version anglaise.

Sample Input 2

```
VVHH
```

Output for Sample Input 2

```
1 2
3 4
```

La version française figure à la suite de la version anglaise.

Problem J5: Rule of Three

Problem Description

A *substitution rule* describes how to take a sequence of symbols and convert it into a different sequence of symbols. For example, ABA $\rightarrow$ BBB, is a substitution rule which means that ABA can be replaced with BBB. Using this rule, the sequence A**ABA**A would be transformed into the sequence ABBBA (the substituted symbols are in **bold**).

In this task, you will be given three substitution rules, a starting sequence of symbols and a final sequence of symbols. You are to use the substitution rules to convert the starting sequence into the final sequence, using a specified number of substitutions.

For example, if the three substitution rules were:

1. AA $\rightarrow$ AB
2. AB $\rightarrow$ BB
3. B $\rightarrow$ AA

we could convert the sequence AB into AAAB in 4 steps, by the following substitutions:

AB $\rightarrow$ **B**B $\rightarrow$ AA**B** $\rightarrow$ AA**AA** $\rightarrow$ AAAB,

where the symbols to be replaced are shown in **bold**. More specifically, from the initial sequence AB, substitute rule 2 starting at position 1, to get the result BB. From BB, substitute rule 3, starting at position 1, to get the result AAB. From AAB, substitute rule 3, starting at position 3, to get the result AAAA. From AAAA, substitute rule 1, starting at position 3, to get the result AAAB, which is the final sequence.

Input Specification

The first three lines will contain the substitution rules. Each substitution rule will be a sequence of `A`'s and `B`'s, followed by a space following by another sequence of `A`'s and `B`'s. Both sequences will have between one and five symbols.

The next line contains three space separated values, S, I and F. The value S ($1 \leq S \leq 15$) is an integer specifying the number of steps that must be used, and the values I (the initial sequence) and F (the final sequence) are sequences of `A`'s and `B`'s, where there are at least one and at most 5 symbols in I and at least one and at most 50 symbols in F.

For 7 of the 15 marks available, $S \leq 6$.

For an additional 7 of the 15 available marks, $S \leq 12$.

Output Specification

The output will be S lines long and describes the substitutions in order.

La version française figure à la suite de la version anglaise.

Line i of the output will contain three space-separated values, R_i, P_i, and W_i:

- R_i is the substitution rule number (either `1`, `2` or `3`) that will be used.
- P_i is the starting position index of where the substitution rule will be applied in the sequence. Notice that the string is 1-indexed (i.e., the first character of the string is at index 1).
- W_i is the sequence that results from this substitution. Specifically, W_i is the sequence of symbols that results by applying substitution rule R_i starting at position P_i from the previous sequence of symbols, W_{i-1}, where we define W_0 to be the initial sequence I. Note that $W_S = F$, the final sequence.

There will always be at least one sequence of S substitutions that will convert I into F. If there is more than one possible sequence of substitutions, any valid sequence will be accepted.

Sample Input

```
AA AB
AB BB
B AA
4 AB AAAB
```

Possible Output for Sample Input

```
2 1 BB
3 1 AAB
3 3 AAAA
1 3 AAAB
```

Explanation of Output for Sample Input

This is the example outlined in the problem description. Note that the following is another possible valid substitution sequence:

```
2 1 BB
3 2 BAA
1 2 BAB
3 1 AAAB
```

Specifically, showing the substitutions in **bold**, we get

AB → B**B** → B**AA** → **B**AB → AAAB.

La version française figure à la suite de la version anglaise.

Problem S2: Pretty Average Primes

Problem Description

For various given positive integers $N > 3$, find two primes, A and B such that N is the average (mean) of A and B. That is, N should be equal to $(A + B)/2$.

Recall that a *prime number* is an integer $P > 1$ which is only divisible by 1 and P. For example, $2,\ 3,\ 5,\ 7,\ 11$ are the first few primes, and $4, 6, 8, 9$ are not prime numbers.

Input Specification

The first line of input is the number T ($1 \leq T \leq 1000$), which is the number of test cases. Each of the next T lines contain one integer N_i ($4 \leq N_i \leq 1\,000\,000$, $1 \leq i \leq T$).

For 6 of the available 15 marks, all $N_i < 1\,000$.

Output Specification

The output will consist of T lines. The ith line of output will contain two integers, A_i and B_i, separated by one space. It should be the case that $N_i = (A_i + B_i)/2$ and that A_i and B_i are prime numbers.

If there are more than one possible A_i and B_i for a particular N_i, output any such pair. The order of the pair A_i and B_i does not matter.

It will be the case that there will always be at least one set of values A_i and B_i for any given N_i.

Sample Input

```
4
8
4
7
21
```

Possible Output for Sample Input

```
3 13
5 3
7 7
13 29
```

Explanation of Possible Output for Sample Input

Notice that:

$$\begin{aligned} 8 &= (3+13)/2, \\ 4 &= (5+3)/2, \\ 7 &= (7+7)/2, \end{aligned}$$

Version française figure à la suite de la version anglaise.

$$21 = (13+29)/2.$$

It is interesting to note, that we can also write

$$\begin{aligned} 8 &= (5+11)/2 \\ 21 &= (5+37)/2 = (11+31)/2 = (19+23)/2 \\ 7 &= (3+11)/2 \end{aligned}$$

and so any of these pairs could have also been used in output. There is no pairs of primes other than 3 and 5 which average to the value of 4.

Footnote
You may have heard about *Goldbach's conjecture*, which states that every even integer greater than 2 can be expressed as the sum of two prime numbers. There is no known proof, yet, so if you want to be famous, prove that conjecture (after you finish the CCC).

This problem can be used to help verify that conjecture, since every even integer can be written as $2N$, and your task is to find two primes A and B such that $2N = A + B$.

Problem S3: Arithmetic Square

Problem Description
You are given a 3×3 grid which contains integers.

Some of the 9 elements in the grid will have a value already, and the remaining elements will be unspecified.

Your task is to determine values for the unspecified elements such that each row, when read from left-to-right is an arithmetic sequence, and that each column, when read from the top-down, is an arithmetic sequence.

Recall that an *arithmetic sequence* of length three is a sequence of integers of the form

$$a, a + d, a + 2d$$

for integer values of a and d. Note that d may be any integer, including zero or a negative integer.

Input Specification
The input will be 3 lines long. Each line will have three space-separated values. Each value will either be an integer in the range from $-1\ 000\ 000$ to $1\ 000\ 000$, inclusive, or the symbol `X`.

For 4 of the 15 marks available, there will be at most 3 `X` symbols in the input.

For an additional 3 of the 15 marks available, all integer values in the input will be between -10 and 10, inclusive.

For an additional 4 of the 15 marks available, there will be at least 7 `X` symbols in the input.

For an additional 2 of the 15 marks available, all integer values in the input will be even numbers.

Output Specification
The output will be 3 lines long. Each line will have three space-separated integers. All integers that were given in the input must be in their same position (i.e., same row and same column as in the input). All rows and columns must form arithmetic sequences. All integers in the output must be between -1 000 000 000 and 1 000 000 000, inclusive.

If there is more than one solution, output any solution. There is guaranteed to be at least one solution.

Sample Input 1
```
8 9 10
16 X 20
24 X 30
```

Output for Sample Input 1
```
8 9 10
```

La version française figure à la suite de la version anglaise.

```
16 18 20
24 27 30
```

Explanation of Output for Sample Input 1

Notice that the second element of the second row must be $16 + t$ and since $20 = 16 + 2t$, then $t = 2$, and thus, this unspecified element must be 18. A similar argument applies to the second element of the third row.

Sample Input 2

```
14 X X
X X 18
X 16 X
```

Possible Output for Sample Input 2

```
14 20 26
18 18 18
22 16 10
```

Explanation of Output for Sample Input 2

This is one of many possible solutions. For example, another solution is:

$$\begin{array}{ccc} 14 & 16 & 18 \\ 14 & 16 & 18 \\ 14 & 16 & 18 \end{array}$$

La version française figure à la suite de la version anglaise.

Problem S4: Tourism

Problem Description
You are planning a trip to visit N tourist attractions. The attractions are numbered from 1 to N and must be visited in this order. You can visit at most K attractions per day, and want to plan the trip to take the fewest number of days as possible.

Under these constraints, you want to find a schedule that has a nice balance between the attractions visited each day. To be precise, we assign a score a_i to attraction i. Given a schedule, each day is given a score equal to the maximum score of all attractions visited that day. Finally, the scores of each day are summed to give the total score of the schedule. What is the maximum possible total score of the schedule, using the fewest days possible?

Input Specification
The first line contains two space-separated integers N and K ($1 \leq K \leq N \leq 10^6$).

The next line contains N space separated integers a_i ($1 \leq a_i \leq 10^9$).

For 3 of the 15 available marks, $2K \geq N$.

For an additional 3 of the 15 available marks, $K \leq 100$ and $N \leq 10^5$.

Output Specification
Output a single integer, the maximum possible total score.

Sample Input
```
5 3
2 5 7 1 4
```

Output for Sample Input
```
12
```

Explanation of Output for Sample Input
We need to have at least two days to visit all the attractions, since we cannot visit all attractions in one day.

Visiting the first two attractions on day 1 will give a score of 5, and visiting the last three attractions on day 2 will give a score of 7, for a total score of 12.

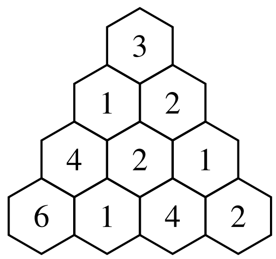
Visiting three attractions on day 1, and two attractions on day 2, which is the only possibility to visit in the fewest number of days possible, would yield a total score of $7 + 4 = 11$.

La version française figure à la suite de la version anglaise.

Problem S5: Triangle: The Data Structure

Problem Description

In a parallel universe, the most important data structure in computer science is the *triangle*. A triangle of size M consists of M rows, with the i-th row containing i elements. Furthermore, these rows must be arranged to form the shape of an equilateral triangular. That is, each row is centered around a vertical line of symmetry through the middle of the triangle. For example, the diagram below shows a triangle of size 4:



A triangle contains sub-triangles. For example, the triangle above contains ten sub-triangles of size 1, six sub-triangles of size 2 (two of which are the triangle containing $(3, 1, 2)$ and the triangle containing $(4, 6, 1)$), three sub-triangles of size 3 (one of which contains $(2, 2, 1, 1, 4, 2)$). Note that every triangle is a sub-triangle of itself.

You are given a triangle of size N and must find the sum of the maximum elements of every sub-triangle of size K.

Input Specification

The first line contains two space-separated integers N and K ($1 \leq K \leq N \leq 3000$).

Following this are N lines describing the triangle. The i-th of these lines contains i space-separated integers $a_{i,j}$ ($0 \leq a_{i,j} \leq 10^9$), representing the i-th row of the triangle.

For 4 of the 15 available marks, $N \leq 1000$.

Output Specification

Output the integer sum of the maximum elements of every sub-triangle of size K.

Sample Input

```
4 2
3
1 2
4 2 1
6 1 4 2
```

Output for Sample Input

```
23
```

La version française figure à la suite de la version anglaise.

2019 Canadian Computing
Problem 1
Human Error

Time Limit: 1 second

Problem Description

Justin and Donald are playing their favourite game: hop chess. You probably haven't heard of it, but the rules are pretty simple.

The board is a rectangular grid, with each square of the board initially having exactly one player's piece on it. Justin's pieces are denoted as `J`, with Donald's being `D`. Each player has at least one piece on the grid initially.

The game begins with Justin playing first. On a turn, a player may move one of his pieces to capture (and thus remove from the board) a neighbouring piece (not necessarily the opponent's). A piece X is said to be neighbouring Y if it is either up, down, left, or right of Y. If such a move cannot be made, then the active player loses.

In an ideal world, both Justin and Donald are perfect logicians, and capable of discerning an optimal strategy for any board. Then perhaps we might be interested in who of the two would win. But that wouldn't be very realistic. Indeed, when playing, Justin and Donald can both come up with a relatively good solution; exactly how good it tends to be is determined by their error factors, J and D respectively.

Formally, the active player with error factor A first chooses a *proposal set*: either the set of all possible moves if there are A or less possible moves or some subset of size A from the set of possible moves if he has more than A possible moves. Then, from this *proposal set*, the player selects a move randomly with equal probability.

Both players, when given the choice of choosing their *proposal set*, chooses the most optimal such set (one which produces the highest probability of winning), knowing that the other player always chooses their *proposal set* optimally as well.

The natural question is then: exactly what is the probability that Justin wins a game of hop chess, given the initial board, J, and D?

Input Specification

Input will begin with two space-separated positive integers R, C $(R \cdot C \leq 13)$. On the next R lines will be strings of C characters drawn from the set $\{$`J, D`$\}$, describing the initial board state. Finally, there will be two space-separated integers, J, D $(1 \leq J, D \leq 13)$

Output Specification

Output a single floating point number rounded to 3 decimal places: the probability that Justin wins.

Sample Input 1

```
1 3
JJD
3 1
```

Output for Sample Input 1

```
0.667
```

Explanation of Output for Sample Input 1

Note that Justin has 3 possible moves (note that _ indicates an empty cell in all explanations below):

- he moves his first piece right, capturing his second piece, and ensuring his loss by having the board appear as `_JD`;
- he moves his second piece right, capturing Donald's piece and securing victory, with the board as `J_J`;
- or he moves his second piece left, capturing his own piece, but leaving Donald unable to move, thus also winning, with the board as `J_D`.

Clearly the latter 2 cases are optimal—but since Justin has error factor 3, there is a 1/3 chance that he chooses the option causing him to lose. Thus he wins with probability 2/3.

Sample Input 2

```
2 2
JJ
DD
3 1
```

Output for Sample Input 2

```
0.000
```

Explanation of Output for Sample Input 2

There is no hope for Justin to win.

To see why, notice that Justin has 4 possible first moves:

```
J_    _J    J_    _J
DD    DD    DJ    JD
```

He can pick any subset of size three from the above moves.

However, Donald will always pick his most optimal move. Regardless of Justin's first move, Donald will leave the board in one of the following configurations:

```
D_    _J    _D    J_
_D    D_    D_    _D
```

all of which will cause Justin to lose.

2019 Canadian Computing
Problem 2
Sirtet

Time Limit: 2 seconds

Problem Description

In a fancy new zero-person video game, Sirtet, the game is a rectangular grid with N rows and M columns. Before the game begins, some grid cells are blank (denoted as `.`) and others are filled (denoted as #). The filled squares represent a set of objects, and the filled squares that are adjacent (horizontally or vertically) should be considered to be part of the same rigid object. For example, this initial grid:

```
..#.
##.#
.##.
#...
#...
```

has four objects, shown below:

```
##      #     #     #
 ##     #
```

When the game begins, the objects fall straight down the grid, all at the same speed. Each object continues to fall straight down until it either touches the bottom row, or has some part of it land directly on top of another object, at which point it stops. What will be the final state of the grid?

Input Specification

The first line contains two space-separated positive integers N and M ($N \cdot M \leq 10^6$).

The following N lines contain M characters each, describing the initial state of the grid. If the j-th column of the i-th row of the grid contains a block, the corresponding character in the input will be a #, otherwise it will be a `.` character.

For 10 of the 25 available marks, $N \cdot M \leq 2000$.

For an additional 6 of the 25 available marks, $M = 2$.

Output Specification

Output N lines contain M characters each, describing the final state of the grid. If the j-th column of the i-th row of the grid contains a block, the corresponding character in the input will be a `#`, otherwise it will be a `.` character.

Sample Input

```
5 4
..#.
##.#
.##.
#...
#...
```

Output for Sample Input

```
....
....
###.
###.
#..#
```

2019 Canadian Computing
Problem 3
Winter Driving

Time Limit: 1 second

Problem Description

In the Great White North, there are N cities numbered from 1 to N. There are A_i citizens living in city i. There are $N - 1$ roads numbered from 2 to N. Road j connects city j and city P_j, where $P_j < j$. There are at most 36 roads connected to any city.

During winter, all roads will be converted into one-way highways due to dangerous driving conditions. That is, road j will become a highway that is either one-way from city j to city P_j or one-way from city P_j to city j.

Every citizen wants to send a holiday card to every other citizen. Citizen x can send a card to citizen y if it is possible to travel from the city x lives in to the city y lives in using only highways.

What is the maximum number of holiday cards that can be sent after converting all roads to highways?

Input Specification

The first line contains one integer N $(2 \le N \le 200\,000)$.

The second line contains N integers $A_1, \cdots, A_N$ $(1 \le A_i \le 10\,000)$.

The third line contains $N - 1$ integers $P_2, \cdots, P_N$ $(1 \le P_j \le j)$.

Let D be the maximum number of roads connected to any city. It is guaranteed that $D \le 36$.

For 5 of the 25 available marks, $N \le 10$.

For an additional 5 of 25 available marks, $N \le 1\,000$ and $D \le 10$.

For an additional 5 of 25 available marks, $D \le 18$.

For an additional 5 of 25 available marks, there will be 37 cities, where one city is connected to 36 other cities, and these other 36 cities are only connected to this one city.

Output Specification

Print one line with one integer, the maximum number of cards that can be sent after converting all roads to highways.

Sample Input

```
4
3 3 4 1
```

```
1 2 1
```

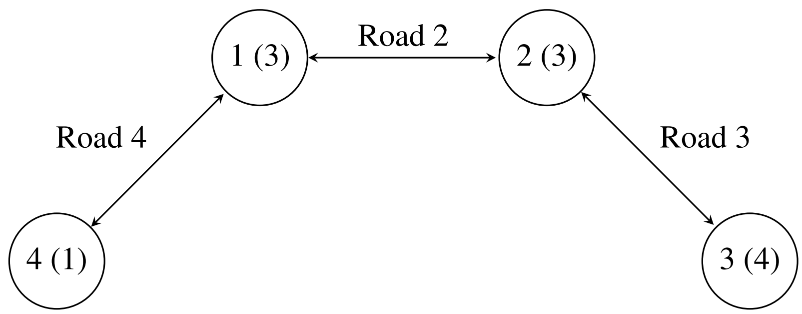
Output for Sample Input

```
67
```

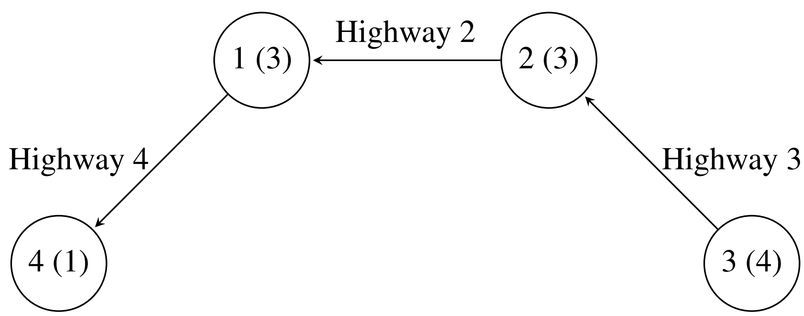
Explanation of Output for Sample Input

One possible way of converting roads to highways is for road 2 to become one-way from city 2 to city 1, road 3 to become one-way from city 3 to city 2, and road 4 to become one-way from city 1 to city 4.

Consider the following pictures, with the cities and associated population (in parentheses) for the initial roads



and what it looks like after all roads are converted to highways:



Every citizen in city 3 can send 3 holiday cards to city 3 citizens, 3 holiday cards to city 2 citizens, 3 holiday cards to city 1 citizens, and 1 holiday card to the city 4 citizen, for a total of 40 holiday cards sent out of city 3.

Similarly,

- city 2 citizens send 6 holidays cards each, for a total of 18 holiday cards.
- city 1 citizens send 3 holidays cards each, for a total of 9 holiday cards.
- the city 4 citizen cannot send any holiday cards.

A total of $40 + 18 + 9 = 67$ holiday cards are sent.

2019 Canadian Computing
Problem 4
Card Scoring

Time Limit: 6 seconds

Problem Description

You have a deck of n cards. Each card has a value: the card values lie between 1 to n, possibly with some repeated values, and possibly with some values never occurring. There is also a special integer k that will be used in calculating a score.

You are playing a game that involves drawing all the cards from the deck one by one. When you draw a card, you may choose to either add it to your hand or discard it. You may also *score* your entire hand at any time. When you score a hand with x cards, you gain $x^{\frac{1}{2}k}$ points and then you discard all cards in that hand. At any point in time, your hand may only contain cards with the same number on them. Given the initial order of the cards in the deck, what is the maximum possible score you can obtain?

Input Specification

The first line of input will contain two space-separated integers k and n. The value k is the value used in the expression $x^{\frac{1}{2}k}$ to compute points ($2 \leq k \leq 4$). The value n is the number of cards in the deck ($1 \leq n \leq 1\ 000\ 000$). The next n lines contain one integer per line, where the ith of these n lines is the i^{th} card from the top of the deck $(1 \leq i \leq n)$.

For 4 of the 25 marks available, $1 \leq n \leq 20$.

For an additional 1 of the 25 marks available, $1 \leq n \leq 300$, $k = 2$.

For an additional 5 of the 25 marks available, $1 \leq n \leq 300$.

For an additional 3 of the 25 marks available, $1 \leq n \leq 5\,000$.

For an additional 3 of the 25 marks available, $k = 4$.

Output Specification

Output one floating point number, which is the maximum score you can obtain from playing optimally.

If your answer is p and the correct answer is q, then your answer will be considered correct if

$$\frac{|p-q|}{q} \leq 10^{-6}.$$

Sample Input 1

```
3 5
1
2
2
1
1
```

Output for Sample Input 1

```
6.656854249
```

Explanation for Output for Sample Input 1

We know the cards we draw in order are $[1, 2, 2, 1, 1]$ and that discarding a hand of x cards gives us a score of $x^{1.5}$.

The optimal strategy is to draw one card, score that hand, draw two cards, score that hand, and draw two more cards and score that hand. This strategy gives a score of $1^{1.5} + 2^{1.5} + 2^{1.5} \approx 6.656854249$.

Sample Input 2

```
4 5
1
2
2
1
1
```

Output for Sample Input 2

```
9.0
```

Explanation of Output for Sample Input 2

We know the cards we draw in order are $[1, 2, 2, 1, 1]$ and that scoring a hand of x cards gives us a score of x^2.

An optimal strategy is to take all cards with 1, and score them all at the end. This strategy gives a score of $3^2 = 9$. Note that taking the first card and scoring, then taking the next two cards and scoring, and then taking the final two cards and scoring, will also yield $1^2 + 2^2 + 2^2 = 9$.

2019 Canadian Computing
Problem 5
Marshmallow Molecules

Time Limit: 4 seconds

Problem Description

Hannah is building a structure made of marshmallows and skewers for her chemistry class. The structure will contain N marshmallows, numbered from 1 to N. Some marshmallows will be connected by skewers. Each skewer connects two marshmallows.

Hannah has M requirements for her structure. Each requirement is given as a pair (a_i, b_i), which means that there must be a skewer connecting marshmallow a_i and marshmallow b_i.

To ensure the stability of the structure, the following requirement must also be satisfied: if $a < b < c$, and if there is a skewer connecting marshmallow a to marshmallow b, and if there is a skewer connecting marshmallow a to marshmallow c, then there must also be a skewer connecting marshmallow b to marshmallow c.

Due to austerity measures imposed by the principal's office, skewers are scarce in Hannah's school. Find the minimum number of skewers necessary to satisfy all requirements.

Input Specification

The first line contains two space-separated integers N and M ($1 \le N, M \le 10^5$).

The next M lines each contain two space-separated integers, with the i-th line containing a_i and b_i ($1 \le a_i < b_i \le N$). All M pairs (a_i, b_i) are distinct.

For 5 of the 25 marks available, $N \le 100$.

For an additional 5 of the 25 marks available, $N \le 5000$.

For an additional 5 of the 25 marks available, for all $1 \le j \le N$, there is at most one pair (a_i, b_i) such that $b_i = j$.

Output Specification

Output a single integer, the minimum total number of skewers.

Sample Input 1

```
6 4
1 2
1 4
4 6
4 5
```

Output for Sample Input 1

```
6
```

Explanation for Output for Sample Input 1

In addition to those already required, there must be skewers between the pairs of marshmallows $(2, 4)$ and $(5, 6)$.

Sample Input 2

```
7 6
2 3
2 6
2 7
1 3
1 4
1 5
```

Output for Sample Input 2

```
16
```

2019 Canadian Computing

Problem 6

Bad Codes

Time Limit: 1 second

Problem Description

Your friend has constructed a code that they want to use to send secret messages to you. The messages will only be composed of N different symbols and each symbol will correspond to one binary sequence with at most M bits.

However, you are not sure the code is going to work: there is a chance that a binary sequence can correspond to two (or more) different messages.

For example, if the code was:

$$\begin{aligned} A &\rightarrow 101 \\ B &\rightarrow 10 \\ C &\rightarrow 1 \\ D &\rightarrow 100 \end{aligned}$$

then the binary sequence 101 could be correspond to either A or BC.

Your job is determine the length of the shortest binary sequence that corresponds to two different messages, or determine that there are no binary sequences which correspond to two different messages.

Input Specification

The first line of input will contain two space-separated integers N and M ($1 \leq N, M \leq 50$). The next N lines of input each will have at least one and at most M characters from the set $\{0, 1\}$.

For 4 of the 25 marks available, $N = 4$ and $M \leq 6$.

For an additional 7 of the 25 marks available, each of the binary sequences contains exactly one `1` bit. For example, the sequences `00100` or `1000` would be valid in this case.

Output Specification

Output will be one line long.

If there is a binary sequence that corresponds to two (or more) messages, print the length of the shortest such binary sequence; otherwise, output one line containing `-1`.

Sample Input 1

```
4 3
101
10
1
100
```

Output for Sample Input 1

```
3
```

Explanation of Output for Sample Input 1

This is the sample in the problem description.

Sample Input 2

```
4 4
1011
1000
1111
1001
```

Output for Sample Input 2

```
-1
```

Explanation of Output for Sample Input 2

There is no binary sequence that corresponds to more than one message. Notice that since each code is 4 bits, and none are the same, every encoding of $4k$ bits can be uniquely decoded into k characters.