Pacific Northwest Regional Programming Contest
Division 2

6 March 2021

- The languages supported are C, C++ 17 (with Gnu extensions), Java, Python 3 (with pypy3), and Kotlin.
- Python 2 and C# are not supported this year.
- For all problems, read the input data from standard input and write the results to standard output.
- In general, when there is more than one integer or word on an input line, they will be separated from each other by exactly one space. No input lines will have leading or trailing spaces, and tabs will never appear in any input.
- Submit only a single source file for each problem.
- Python may not have sufficient performance for many of the problems; use it at your discretion.

Problem R
No Thanks!

Time Limit: 1

In the card game “No Thanks,” the deck of cards consists of 36 cards numbered 1–36, and players collect cards to their score pile as the game is played. A player’s final score is the sum of the numbers on their collected cards, with one exception: if a player has collected any cards with two or more consecutive numbers, only the smallest number of that group counts toward the score. Your job is to compute the score for a single player’s pile of cards, though here we allow play with a deck much larger than 36 cards.

Input

The first line contains one integer, n, representing the number of cards collected. The second line contains n integers representing the numbers on the collected cards. You may assume that $1 \leq n \leq 90\ 000$, all card values are in the range $1 \ldots 90\ 000$ inclusive, and no card value is repeated.

Output

Output a single line containing the score for the given set of cards.

Sample Input 1	Sample Output 1
5 1 7 5 3 4	11

Sample Input 2	Sample Output 2
6 2 1 3 8 4 5	9

Problem S
Exam Manipulation
Time Limit: 1

A group of students is taking a True/False exam. Each question is worth one point. You, as their teacher, want to make your students look as good as possible—so you cheat! (I know, you would never actually do that.) To cheat, you manipulate the answer key so that the lowest score in the class is as high as possible.

What is the best possible lowest score you can achieve?

Input

The first line of input contains two integers n $(1 \le n \le 1{,}000)$ and k $(1 \le k \le 10)$, where n is the number of students, and k is the number of True/False questions on the exam.

Each of the next n lines contains a string of length k, consisting only of upper-case 'T' and upper-case 'F'. This string represents the answers that a student submitted, in the order the questions were given.

Output

Output, on a single line, the best possible lowest score in the class.

Sample Input 1	Sample Output 1
5 4 TFTF TFFF TFTT TFFT TFTF	2

Sample Input 2	Sample Output 2
3 5 TFTFT TFTFT TFTFT	5

Problem T
Painted Corridors

Time Limit: 5

The Institute of Colorfully Painted Corridors is planning the construction of a new building. The building has numerous junctions, and corridors that each connect a pair of junctions. The corridors will be painted by amazing new painting robots that drive along the corridors and paint all the walls as they go. The architect has specified the colors of some of the corridors, which may be red, orange, yellow, green, blue, or purple. However, there is only a budget for three painting robots, so there will be a single robot for each primary color (red, yellow, or blue). In addition, these robots are the cheapest possible version, and cannot turn their paint sprayer off (though they can go as fast or as slow as desired with no problems; they can even stop moving entirely).

If a corridor needs to be painted a secondary color (orange, green, or purple), in order for the paints to mix properly, the two robots with the appropriate primary colors must travel down the corridor in the same direction at the same time to create the correct color. The color mixing rules are: $orange = red + yellow$, $green = yellow + blue$, and $purple = red + blue$. A corridor that is unspecified in the plan may be painted any color, or left unpainted.

Corridors may be painted multiple times, provided that each time they are painted with the correct color. Corridors with no specified color can be painted multiple times with different colors. All corridors can be travelled along in both directions. The robots may end up at any junctions after painting all the corridors.

Given the architect's design, is it possible for the painting robots to paint the corridors the desired colors?

Input

The first line of input contains five integers, n ($2 \leq n \leq 100$), m ($1 \leq m \leq \frac{n \cdot (n-1)}{2}$), r, b and y ($1 \leq r, b, y \leq n$), where n is the number of junctions, m is the number of corridors, and r, b and y are the initial junctions of the red, blue, and yellow painting robots respectively. Junctions are numbered 1 through n. Each of the next m lines contains two integers i, j ($1 \leq i < j \leq n$), and a single character c which is one of `R`, `O`, `Y`, `G`, `B`, `P`, `X`. The integers i, j indicate that there is a corridor between junction i and junction j, with c indicating the desired color. (`R`, `O`, `Y`, `G`, `B`, `P`, `X`, corresponding to Red, Orange, Yellow, Green, Blue, Purple, and Unspecified, respectively.) There is at most one corridor between each pair of junctions.

Output

Output a single integer, 1 if it is possible to paint the corridors as described and 0 otherwise.

Sample Input 1	Sample Output 1
6 5 1 2 5 1 3 X 2 3 X 3 4 P 4 5 X 4 6 Y	1

Sample Input 2	Sample Output 2
6 5 1 2 5 1 3 X 2 3 X 3 4 O 4 5 X 4 6 Y	0

Sample Input 3	Sample Output 3
6 5 1 2 5 1 3 X 2 3 X 3 4 P 4 5 X 4 6 G	1

Problem U
Basic Basis
Time Limit: 2

You are given a sequence of n bit strings $b_1, b_2, \ldots, b_n$, each with $k \times 4$ bits.

You are also given another sequence of m bit strings $a_1, a_2, \ldots, a_m$, each also with $k \times 4$ bits.

Let $f(x)$ denote the minimum index i such that it is possible to take a non-empty subset of $b_1, b_2, \ldots, b_i$, XOR them all together, and get x. If there is no such index, $f(x) = -1$.

Print the values $f(a_1), f(a_2), \ldots, f(a_m)$.

Input

The first line of input contains three integers n ($1 \le n \le 1{,}000$), m ($1 \le m \le 1{,}000$) and k ($1 \le k \le 40$), where n is the length of sequence b, m is the length of sequence a, and the elements of both sequences are bit strings with $k \times 4$ bits.

Each of the next n lines contains a hexadecimal representation of b_i as a string of length k. The strings consist only of hexadecimal digits (‘0’–‘9’ and ‘a’–‘f’).

Then, each of the next m lines contains a hexadecimal representation of a_i in the same format as above.

Output

Output m lines with a single integer on each line, where the integer on the ith line is $f(a_i)$.

Sample Input 1	Sample Output 1
3 5 2 02 e1 fa 02 e3 1b e1 ff	1 2 3 2 -1

Sample Input 2	Sample Output 2
5 6 2 01 02 04 08 10 01 02 03 04 05 64	1 2 2 3 3 -1

Problem V
Bitonic Ordering

Time Limit: 2

Noah suggests the following card game: You are given a deck of cards, each with a distinct positive integer value written on it. The cards are shuffled and placed in a row. Your objective is to arrange the cards in the row so that the values are monotonically increasing initially, and then monotonically decreasing for the remainder of the sequence.

The only move that is allowed is that two neighboring cards may swap positions. Cards may only swap positions if they are adjacent to each other.

Note that in the final ordered sequence, the initial increasing portion of the sequence may be empty (such that the whole sequence is in descending order). Likewise it is allowed for the decreasing portion of the sequence to be empty.

What is the fewest number of moves needed to get the cards arranged in the proper order?

Input

The first line of input contains a single integer n ($1 \le n \le 3 \cdot 10^5$), which is the number of cards.

Each of the next n lines contains a single integer c ($1 \le c \le 10^9$). These are the cards, in their initial order. They will all be distinct.

Output

Output a single integer, which is the fewest number of moves needed to arrange the cards as specified.

Sample Input 1	Sample Output 1
8 7 4 8 10 1 2 6 9	7

Problem W
Derangement Rotations
Time Limit: 1

A *Derangement* is a permutation p of $1, 2, \ldots, n$ where $p_i \neq i$ for all i from 1 to n.

A *rotation* of a sequence $a_1, a_2, \ldots, a_n$ with offset k ($1 \leq k \leq n$) is equal to the sequence a_k, a_{k+1}, . . ., a_n, a_1, a_2, . . ., a_{k-1}. A sequence of length n has at most n distinct rotations.

Given a derangement D, let $f(D)$ denote the number of distinct rotations of D that are also derangements. For example, $f([2, 1]) = 1$, $f([3, 1, 2]) = 2$.

Given n and a prime number p, count the number of derangements D of $1, 2, \ldots, n$ such that $f(D) = n - 2$, modulo p.

Input

The single line of input contains two integers n ($3 \leq n \leq 10^6$) and p ($10^8 \leq p \leq 10^9 + 7$), where n is a permutation size, and p is a prime number.

Output

Output a single integer, which is the number of derangements D of size n with $f(D) = n - 2$, modulo p.

Sample Input 1	Sample Output 1
`3 1000000007`	`0`

Sample Input 2	Sample Output 2
`6 999999937`	`20`

Problem X
Ant Typing
Time Limit: 1

Consider a configurable keyboard where keys can be moved about. An ant is walking on the top row of this keyboard and needs to type a numeric string. The ant starts on the leftmost key of the top row, which contains 9 keys, some permutation of the digits from 1 to 9. On a given second, the ant can perform one of three operations:

1. Stay on that key. The digit corresponding to that key will be entered.
2. Move one key to the left. This can only happen if the ant is not on the leftmost key.
3. Move one key to the right. This can only happen if the ant is not on the rightmost key.

Compute the minimum number of seconds needed for the ant to type out the given numeric string, over all possible numeric key permutations.

Input

The single line of input contains a single string s ($1 \leq |s| \leq 10^5$) consisting only of numeric digit characters from 1 to 9. This is the numeric string that the ant needs to type.

Output

Output a single integer, which is the minimum number of seconds needed for the ant to type out the given numeric string, over all possible numeric key permutations.

Sample Input 1	Sample Output 1
78432579	20

Problem Y
Dominating Duos

Time Limit: 4

A group of people are standing in a line. Each person has a distinct height. You would like to count the number of unordered pairs of people in the line such that they are taller than everyone in between them in the line.

More formally, let d be a sequence of the heights of the people in order from left to right. We want to count the number of pairs of indices i and j with $i < j$ such that for all k with $i < k < j$, $d_i > d_k$ and $d_j > d_k$. Note that if $j = i + 1$ (*i.e.*, there are no k's between i and j), it is trivially true.

Input

The first line of input contains an integer n ($2 \leq n \leq 10^6$), which is the number of people.

Each of the next n lines contains a single integer d_i ($1 \leq d_i \leq n$). These are the heights of the people in the group, in the order in which they're standing. The sequence is guaranteed to be a permutation of the integers 1 through n.

Output

Output a single integer, which is the number of pairs of people who are taller than everyone between them.

Sample Input 1	Sample Output 1
3 2 1 3	3

Sample Input 2	Sample Output 2
6 1 3 2 6 4 5	7

Problem Z
Missing Number

Time Limit: 1

You are teaching kindergarten! You wrote down the numbers from 1 to n, in order, on a whiteboard. When you weren't paying attention, one of your students erased one of the numbers.

Can you tell which number your mischievous student erased?

Input

The first line of input contains a single integer n ($2 \le n \le 100$), which is the number of numbers that you wrote down.

The second line of input contains a string of digits, which represents the numbers you wrote down (minus the one that has been erased). There are no spaces in this string. It is guaranteed to contain all of the numbers from 1 to n, in order, except for the single number that the student erased.

Output

Output a single integer, which is the number that the tricky student erased.

Sample Input 1	Sample Output 1
5 1235	4

Sample Input 2	Sample Output 2
10 1234568910	7

Sample Input 3	Sample Output 3
15 1234567891012131415	11

Problem AA
Longest Common Subsequence

Time Limit: 1

You are given n strings, each a permutation of the first k upper-case letters of the alphabet.

String s is a *subsequence* of string t if and only if it is possible to delete some (possibly zero) characters from the string t to get the string s.

Compute the length of the longest common *subsequence* of all n strings.

Input

The first line of input contains two integers n ($1 \leq n \leq 10^5$) and k ($1 \leq k \leq 26$), where n is the number of strings, and the strings are all permutations of the first k upper-case letters of the alphabet.

Each of the next n lines contains a single string t. It is guaranteed that every t contains each of the first k upper-case letters of the alphabet exactly once.

Output

Output a single integer, the length of the longest subsequence that appears in all n strings.

Sample Input 1	Sample Output 1
2 3 BAC ABC	2

Sample Input 2	Sample Output 2
3 8 HGBDFCAE ADBGHFCE HCFGBDAE	3

Sample Input 3	Sample Output 3
6 8 AHFBGDCE FABGCEHD AHDGFBCE DABHGCFE ABCHFEDG DGABHFCE	4

Problem AB
Rating Problems

Time Limit: 1

Your judges are preparing a problem set, and they're trying to evaluate a problem for inclusion in the set. Each judge rates the problem with an integer between -3 and 3, where:

3	means: I *really* like this problem!
-3	means: I *really don't* like this problem!
0	means: Meh. I don't care if we use this problem or not.

The overall rating of the problem is the average of all of the judges' ratings—that is, the sum of the ratings divided by the number of judges providing a rating.

Some judges have already rated the problem. Compute the minimum and maximum possible overall rating that the problem can end up with after the other judges submit their ratings.

Input

The first line of input contains two integers n ($1 \leq n \leq 10$) and k ($0 \leq k \leq n$), where n is the total number of judges, and k is the number of judges who have already rated the problem.

Each of the next k lines contains a single integer r ($-3 \leq r \leq 3$). These are the ratings of the k judges that have already rated the problem.

Output

Output two space-separated floating point numbers on a single line, which are the minimum and maximum overall rating the problem could achieve after the remaining judges rate the problem, minimum first. These values must be accurate to an absolute or relative error of 10^{-4}.

Sample Input 1	Sample Output 1
5 2 1 2	-1.2 2.4

Sample Input 2	Sample Output 2
4 4 -3 -3 -2 -3	-2.75 -2.75

Problem AC
Magic Trick
Time Limit: 1

You are performing a magic trick with a special deck of cards.

You lay out the cards in a row from left to right, face up. Each card has a lower-case letter on it. Two cards with the same letter are indistinguishable. You select an audience member to perform an operation on the cards. You will not see what operation they perform.

The audience member can do one of two things—they can either select any two cards and swap them, or leave the cards untouched.

In order for the trick to succeed, you must correctly guess what the audience member did—either you guess that the audience member did nothing, or you point at the two cards the audience member swapped.

Given a string that represents the initial arrangement of the cards, can you guarantee that you will always be able to guess the audience member's operation correctly, no matter what operation they perform?

Input

The input consists of a single line containing the string s ($1 \leq |s| \leq 50$), which represents the initial arrangement of the cards, in the order they appear in the row. The string contains only lower-case letters ('`a`'–'`z`').

Output

Output a single line with 1 if you can guarantee that you will always be able to guess the audience member's operation correctly, or 0 otherwise.

Sample Input 1	Sample Output 1
`robust`	`1`

Sample Input 2	Sample Output 2
`icpc`	`0`

Problem AD
Bad Packing
Time Limit: 6

We have a knapsack of integral capacity and some objects of assorted integral sizes. We attempt to fill the knapsack up, but unfortunately, we are really bad at it, so we end up wasting a lot of space that can't be further filled by any of the remaining objects. In fact, we are optimally bad at this! How bad can we possibly be?

Figure out the least capacity we can use where we cannot place any of the remaining objects in the knapsack. For example, suppose we have 3 objects with weights 3, 5 and 3, and our knapsack has capacity 6. If we foolishly pack the object with weight 5 first, we cannot place either of the other two objects in the knapsack. That's the worst we can do, so 5 is the answer.

Input

The first line of input contains two integers n ($1 \leq n \leq 1{,}000$) and c ($1 \leq c \leq 10^5$), where n is the number of objects we want to pack and c is the capacity of the knapsack.

Each of the next n lines contains a single integer w ($1 \leq w \leq c$). These are the weights of the objects.

Output

Output a single integer, which is the least capacity we can use where we cannot place any of the remaining objects in the knapsack.

Sample Input 1	Sample Output 1
3 6 3 5 3	5

Problem AE
Kangaroo Party
Time Limit: 1

A group of kangaroos live in houses on the number line. They all want to watch the Kangaroo Bowl!

Because not all of the kangaroos can fit a single house, they will designate two kangaroos to each host a party at their house. All other kangaroos will choose to go to the house that is closest to them, picking arbitrarily if they are the same distance from both.

A kangaroo expends $(a - b)^2$ units of energy to travel from location a to location b. Compute the minimum total units of energy expended if the two party house locations are chosen optimally.

Input

The first line of input contains a single integer n ($2 \le n \le 50$), which is the number of kangaroos.

Each of the next n lines contains a single integer x ($-1{,}000 \le x \le 1{,}000$), which is the location on the number line of the house of one of the kangaroos. Each location will be distinct.

Output

Output, on a single line, the minimum total units of energy expended by all the kangaroos, given that the party house locations are chosen optimally.

Sample Input 1	Sample Output 1
5 0 3 -3 10 11	19

Problem AF
Rainbow Numbers
Time Limit: 1

Define a *rainbow number* as an integer that, when represented in base 10 with no leading zeros, has no two adjacent digits the same.

Given lower and upper bounds, count the number of rainbow numbers between them (inclusive).

Input

The first line of input contains a single integer L ($1 \le L < 10^{10^5}$), which is the lower bound.

The second line of input contains a single integer U ($1 \le U < 10^{10^5}$), which is the upper bound.

It is guaranteed that $L \le U$. Note that the limits are not a misprint; L and U can be up to 10^5 digits long.

Output

Output a single integer, which is the number of rainbow numbers between L and U (inclusive). Because this number may be very large, output it modulo $998{,}244{,}353$.

Sample Input 1	Sample Output 1
1 10	10

Sample Input 2	Sample Output 2
12345 65432	35882

Problem AG
Reconstruct Sum

Time Limit: 1

On a whiteboard, you have found a list of integers. Is it possible to use all of them to write down a correct arithmetic expression where one of them is the sum of all the others?

You may not alter the integers in any way (*e.g.*, changing the sign or concatenating).

Input

The first line of input contains an integer n ($1 \leq n \leq 10^4$), representing the number of integers on the whiteboard.

The integers on the whiteboard are given over the next n lines, one per line. Their absolute values are guaranteed to be at most 10^5.

Output

Print a single integer x which is one of the inputs, and is the sum of all the others. If there's more than one such x, output any one. If there are no such values of x, output the string 'BAD'.

Sample Input 1	Sample Output 1
4 1 6 3 2	6

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

Sample Input 3	Sample Output 3
5 1 10 4 2 -3	BAD

Problem AH
Triangular Collection

Time Limit: 1

Call a set of positive integers *triangular* if it has size at least three and, for all triples of distinct integers from the set, a triangle with those three integers as side lengths can be constructed.

Given a set of positive integers, compute the number of its *triangular* subsets.

Input

The first line of input contains a single integer n ($1 \le n \le 50$), which is the number of integers in the set.

Each of the the next n lines contains a single integer x ($1 \le x \le 10^9$). These are the elements of the set. They are guaranteed to be distinct.

Output

Output a single integer, which is the number of triangular subsets of the given set.

Sample Input 1	Sample Output 1
5 3 1 5 9 10	2

Sample Input 2	Sample Output 2
10 27 26 17 10 2 14 1 12 23 39	58