Educational Codeforces Round 39 (Rated for Div. 2)

A. Partition

1 second, 256 megabytes

You are given a sequence a consisting of n integers. You may partition this sequence into two sequences b and c in such a way that every element belongs exactly to one of these sequences.

Let B be the sum of elements belonging to b, and C be the sum of elements belonging to c (if some of these sequences is empty, then its sum is 0). What is the maximum possible value of $B - C$?

Input

The first line contains one integer n ($1 \le n \le 100$) — the number of elements in a.

The second line contains n integers $a_1, a_2, ..., a_n$ ($-100 \le a_i \le 100$) — the elements of sequence a.

Output

Print the maximum possible value of $B - C$, where B is the sum of elements of sequence b, and C is the sum of elements of sequence c.

input

```
3
1 -2 0
```

output

```
3
```

input

```
6
16 23 16 15 42 8
```

output

```
120
```

In the first example we may choose $b = \{1, 0\}$, $c = \{-2\}$. Then $B = 1$, $C = -2$, $B - C = 3$.

In the second example we choose $b = \{16, 23, 16, 15, 42, 8\}$, $c = \{\}$ (an empty sequence). Then $B = 120$, $C = 0$, $B - C = 120$.

B. Weird Subtraction Process

1 second, 256 megabytes

You have two variables a and b. Consider the following sequence of actions performed with these variables:

1. If $a = 0$ or $b = 0$, end the process. Otherwise, go to step 2;
2. If $a \ge 2 \cdot b$, then set the value of a to $a - 2 \cdot b$, and repeat step 1. Otherwise, go to step 3;
3. If $b \ge 2 \cdot a$, then set the value of b to $b - 2 \cdot a$, and repeat step 1. Otherwise, end the process.

Initially the values of a and b are positive integers, and so the process will be finite.

You have to determine the values of a and b after the process ends.

Input

The only line of the input contains two integers n and m ($1 \le n, m \le 10^{18}$). n is the initial value of variable a, and m is the initial value of variable b.

Output

Print two integers — the values of a and b after the end of the process.

input

```
12 5
```

output

```
0 1
```

input

```
31 12
```

output

```
7 12
```

Explanations to the samples:

1. $a = 12, b = 5 \rightarrow a = 2, b = 5 \rightarrow a = 2, b = 1 \rightarrow a = 0, b = 1$;
2. $a = 31, b = 12 \rightarrow a = 7, b = 12$.

C. String Transformation

1 second, 256 megabytes

You are given a string s consisting of $|s|$ small english letters.

In one move you can replace any character of this string to the next character in alphabetical order (`a` will be replaced with `b`, `s` will be replaced with `t`, etc.). You cannot replace letter `z` with any other letter.

Your target is to make some number of moves (not necessary minimal) to get string `abcdefghijklmnopqrstuvwxyz` (english alphabet) as a subsequence. Subsequence of the string is the string that is obtained by deleting characters at some positions. You need to print the string that will be obtained from the given string and will be contain english alphabet as a subsequence or say that it is impossible.

Input

The only one line of the input consisting of the string s consisting of $|s|$ ($1 \le |s| \le 10^5$) small english letters.

Output

If you can get a string that can be obtained from the given string and will contain english alphabet as a subsequence, print it. Otherwise print «-1» (without quotes).

input
```
aacceeggiikkmmooqqssuuwwyy
```
output
```
abcdefghijklmnopqrstuvwxyz
```

input
```
thereisnoanswer
```
output
```
-1
```

D. Timetable

2 seconds, 256 megabytes

Ivan is a student at Berland State University (BSU). There are n days in Berland week, and each of these days Ivan might have some classes at the university.

There are m working hours during each Berland day, and each lesson at the university lasts exactly one hour. If at some day Ivan's first lesson is during i-th hour, and last lesson is during j-th hour, then he spends $j - i + 1$ hours in the university during this day. If there are no lessons during some day, then Ivan stays at home and therefore spends 0 hours in the university.

Ivan doesn't like to spend a lot of time in the university, so he has decided to skip some lessons. He cannot skip more than k lessons during the week. After deciding which lessons he should skip and which he should attend, every day Ivan will enter the university right before the start of the first lesson he does not skip, and leave it after the end of the last lesson he decides to attend. If Ivan skips all lessons during some day, he doesn't go to the university that day at all.

Given n, m, k and Ivan's timetable, can you determine the minimum number of hours he has to spend in the university during one week, if he cannot skip more than k lessons?

Input

The first line contains three integers n, m and k ($1 \le n, m \le 500$, $0 \le k \le 500$) — the number of days in the Berland week, the number of working hours during each day, and the number of lessons Ivan can skip, respectively.

Then n lines follow, i-th line containing a binary string of m characters. If j-th character in i-th line is `1`, then Ivan has a lesson on i-th day during j-th hour (if it is `0`, there is no such lesson).

Output

Print the minimum number of hours Ivan has to spend in the university during the week if he skips not more than k lessons.

input
```
2 5 1
01001
10110
```
output
```
5
```

input
```
2 5 0
01001
10110
```
output
```
8
```

In the first example Ivan can skip any of two lessons during the first day, so he spends 1 hour during the first day and 4 hours during the second day.

In the second example Ivan can't skip any lessons, so he spends 4 hours every day.

E. Largest Beautiful Number

1 second, 256 megabytes

Yes, that's another problem with definition of "beautiful" numbers.

Let's call a positive integer x *beautiful* if its decimal representation without leading zeroes contains even number of digits, and there exists a permutation of this representation which is palindromic. For example, 4242 is a beautiful number, since it contains 4 digits, and there exists a palindromic permutation 2442.

Given a positive integer s, find the largest beautiful number which is less than s.

Input

The first line contains one integer t ($1 \le t \le 10^5$) — the number of testcases you have to solve.

Then t lines follow, each representing one testcase and containing one string which is the decimal representation of number s. It is guaranteed that this string has even length, contains no leading zeroes, and there exists at least one beautiful number less than s.

The sum of lengths of s over all testcases doesn't exceed $2 \cdot 10^5$.

Output

For each testcase print one line containing the largest beautiful number which is less than s (it is guaranteed that the answer exists).

input
```
4
89
88
1000
28923845
```

output

```
88
77
99
28923839
```

F. Fibonacci String Subsequences

3.5 seconds, 256 megabytes

You are given a binary string s (each character of this string is either `0` or `1`).

Let's denote the cost of string t as the number of occurences of s in t. For example, if s is `11` and t is `111011`, then the cost of t is 3.

Let's also denote the Fibonacci strings sequence as follows:

- $F(0)$ is `0`;
- $F(1)$ is `1`;
- $F(i) = F(i-1) + F(i-2)$ if $i > 1$, where $+$ means the concatenation of two strings.

Your task is to calculate the sum of costs of all subsequences of the string $F(x)$. Since answer may be large, calculate it modulo $10^9 + 7$.

Input

The first line contains two integers n and x ($1 \le n \le 100$, $0 \le x \le 100$) — the length of s and the index of a Fibonacci string you are interested in, respectively.

The second line contains s — a string consisting of n characters. Each of these characters is either `0` or `1`.

Output

Print the only integer — the sum of costs of all subsequences of the string $F(x)$, taken modulo $10^9 + 7$.

input

```
2 4
11
```

output

```
14
```

input

```
10 100
1010101010
```

output

```
553403224
```

G. Almost Increasing Array

3 seconds, 512 megabytes

We call an array *almost increasing* if we can erase not more than one element from it so that the array becomes strictly increasing (that is, every element is striclty greater than every element before it).

You are given an array a consisting of n elements. You are allowed to replace any element with any integer number (and you may do so any number of times you need). What is the minimum number of replacements you have to perform in order to make the array *almost increasing*?

Input

The first line contains one integer n ($2 \le n \le 200000$) — the number of elements in a.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the array a.

Output

Print the minimum number of replaces you have to perform so that a is *almost increasing*.

input

```
5
5 4 3 2 1
```

output

```
3
```

input

```
5
1 2 8 9 5
```

output

```
0
```