CODEFORCES β
Sponsored by Telegram

Codeforces Round #604 (Div. 2)

A. Beautiful String

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

A string is called beautiful if no two consecutive characters are equal. For example, "ababcb", "a" and "abab" are beautiful strings, while "aaaaaa", "abaa" and "bb" are not.

Ahcl wants to construct a beautiful string. He has a string s, consisting of only characters 'a', 'b', 'c' and '?'. Ahcl needs to replace each character '?' with one of the three characters 'a', 'b' or 'c', such that the resulting string is beautiful. Please help him!

More formally, after replacing all characters '?', the condition $s_i \neq s_{i+1}$ should be satisfied for all $1 \leq i \leq |s| - 1$, where $|s|$ is the length of the string s.

Input

The first line contains positive integer t $(1 \leq t \leq 1000)$ — the number of test cases. Next t lines contain the descriptions of test cases.

Each line contains a non-empty string s consisting of only characters 'a', 'b', 'c' and '?'.

It is guaranteed that in each test case a string s has at least one character '?'. The sum of lengths of strings s in all test cases does not exceed 10^5.

Output

For each test case given in the input print the answer in the following format:

- If it is impossible to create a beautiful string, print "-1" (without quotes);
- Otherwise, print the resulting beautiful string after replacing all '?' characters. If there are multiple answers, you can print any of them.

Example

input

```
3
a???cb
a??bbc
a?b?c
```

output

```
ababcb
-1
acbac
```

Note

In the first test case, all possible correct answers are "ababcb", "abcacb", "abcbcb", "acabcb" and "acbacb". The two answers "abcbab" and "abaabc" are incorrect, because you can replace only '?' characters and the resulting string must be beautiful.

In the second test case, it is impossible to create a beautiful string, because the 4-th and 5-th characters will be always equal.

In the third test case, the only answer is "acbac".

B. Beautiful Numbers

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a permutation $p = [p_1, p_2, \ldots, p_n]$ of integers from 1 to n. Let's call the number m $(1 \leq m \leq n)$ *beautiful*, if there exists two indices l, r $(1 \leq l \leq r \leq n)$, such that the numbers $[p_l, p_{l+1}, \ldots, p_r]$ is a permutation of numbers $1, 2, \ldots, m$.

For example, let $p = [4, 5, 1, 3, 2, 6]$. In this case, the numbers $1, 3, 5, 6$ are beautiful and $2, 4$ are not. It is because:

- if $l = 3$ and $r = 3$ we will have a permutation $[1]$ for $m = 1$;
- if $l = 3$ and $r = 5$ we will have a permutation $[1, 3, 2]$ for $m = 3$;
- if $l = 1$ and $r = 5$ we will have a permutation $[4, 5, 1, 3, 2]$ for $m = 5$;
- if $l = 1$ and $r = 6$ we will have a permutation $[4, 5, 1, 3, 2, 6]$ for $m = 6$;
- it is impossible to take some l and r, such that $[p_l, p_{l+1}, \ldots, p_r]$ is a permutation of numbers $1, 2, \ldots, m$ for $m = 2$ and for

$m = 4$.

You are given a permutation $p = [p_1, p_2, \dots, p_n]$. For all m ($1 \le m \le n$) determine if it is a beautiful number or not.

Input

The first line contains the only integer t ($1 \le t \le 1000$) — the number of test cases in the input. The next lines contain the description of test cases.

The first line of a test case contains a number n ($1 \le n \le 2 \cdot 10^5$) — the length of the given permutation p. The next line contains n integers $p_1, p_2, \dots, p_n$ ($1 \le p_i \le n$, all p_i are different) — the given permutation p.

It is guaranteed, that the sum of n from all test cases in the input doesn't exceed $2 \cdot 10^5$.

Output

Print t lines — the answers to test cases in the order they are given in the input.

The answer to a test case is the string of length n, there the i-th character is equal to 1 if i is a beautiful number and is equal to 0 if i is not a beautiful number.

Example

input

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

output

```
101011
11111
1001
```

Note

The first test case is described in the problem statement.

In the second test case all numbers from 1 to 5 are beautiful:

- if $l = 3$ and $r = 3$ we will have a permutation $[1]$ for $m = 1$;
- if $l = 3$ and $r = 4$ we will have a permutation $[1, 2]$ for $m = 2$;
- if $l = 2$ and $r = 4$ we will have a permutation $[3, 1, 2]$ for $m = 3$;
- if $l = 2$ and $r = 5$ we will have a permutation $[3, 1, 2, 4]$ for $m = 4$;
- if $l = 1$ and $r = 5$ we will have a permutation $[5, 3, 1, 2, 4]$ for $m = 5$.

C. Beautiful Regional Contest

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

So the Beautiful Regional Contest (BeRC) has come to an end! n students took part in the contest. The final standings are already known: the participant in the i-th place solved p_i problems. Since the participants are primarily sorted by the number of solved problems, then $p_1 \ge p_2 \ge \dots \ge p_n$.

Help the jury distribute the gold, silver and bronze medals. Let their numbers be g, s and b, respectively. Here is a list of requirements from the rules, which all must be satisfied:

- for each of the three types of medals, at least one medal must be awarded (that is, $g > 0$, $s > 0$ and $b > 0$);
- the number of gold medals must be strictly less than the number of silver and the number of bronze (that is, $g < s$ and $g < b$, but there are no requirements between s and b);
- each gold medalist must solve strictly more problems than any awarded with a silver medal;
- each silver medalist must solve strictly more problems than any awarded a bronze medal;
- each bronze medalist must solve strictly more problems than any participant not awarded a medal;
- the total number of medalists $g + s + b$ should not exceed half of all participants (for example, if $n = 21$, then you can award a maximum of 10 participants, and if $n = 26$, then you can award a maximum of 13 participants).

The jury wants to reward with medals the total **maximal** number participants (i.e. to maximize $g + s + b$) so that all of the items listed above are fulfilled. Help the jury find such a way to award medals.

Input

The first line of the input contains an integer t ($1 \le t \le 10000$) — the number of test cases in the input. Then t test cases follow.

The first line of a test case contains an integer n ($1 \le n \le 4 \cdot 10^5$) — the number of BeRC participants. The second line of a test case contains integers $p_1, p_2, \dots, p_n$ ($0 \le p_i \le 10^6$), where p_i is equal to the number of problems solved by the i-th participant from the final standings. The values p_i are sorted in non-increasing order, i.e. $p_1 \geq p_2 \geq \cdots \geq p_n$.

The sum of n over all test cases in the input does not exceed $4 \cdot 10^5$.

Output

Print t lines, the j-th line should contain the answer to the j-th test case.

The answer consists of three non-negative integers g, s, b.

- Print $g = s = b = 0$ if there is no way to reward participants with medals so that all requirements from the statement are satisfied at the same time.
- Otherwise, print three positive numbers g, s, b — the possible number of gold, silver and bronze medals, respectively. The sum of $g + s + b$ should be the maximum possible. If there are several answers, print any of them.

Example

input

```
5
12
5 4 4 3 2 2 1 1 1 1 1 1
4
4 3 2 1
1
1000000
20
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
32
64 64 63 58 58 58 58 58 37 37 37 37 34 34 28 28 28 28 28 28 24 24 19 17 17 17 17 16 16 16 16 11
```

output

```
1 2 3
0 0 0
0 0 0
2 5 3
2 6 6
```

Note

In the first test case, it is possible to reward 1 gold, 2 silver and 3 bronze medals. In this case, the participant solved 5 tasks will be rewarded with the gold medal, participants solved 4 tasks will be rewarded with silver medals, participants solved 2 or 3 tasks will be rewarded with bronze medals. Participants solved exactly 1 task won't be rewarded. It's easy to see, that in this case, all conditions are satisfied and it is possible to reward participants in this way. It is impossible to give more than 6 medals because the number of medals should not exceed half of the number of participants. The answer 1, 3, 2 is also correct in this test case.

In the second and third test cases, it is impossible to reward medals, because at least one medal of each type should be given, but the number of medals should not exceed half of the number of participants.

D. Beautiful Sequence

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

An integer sequence is called *beautiful* if the difference between any two consecutive numbers is equal to 1. More formally, a sequence $s_1, s_2, \ldots, s_n$ is beautiful if $|s_i - s_{i+1}| = 1$ for all $1 \leq i \leq n-1$.

Trans has a numbers 0, b numbers 1, c numbers 2 and d numbers 3. He wants to construct a beautiful sequence using all of these $a + b + c + d$ numbers.

However, it turns out to be a non-trivial task, and Trans was not able to do it. Could you please help Trans?

Input

The only input line contains four non-negative integers a, b, c and d ($0 < a + b + c + d \leq 10^5$).

Output

If it is impossible to construct a beautiful sequence satisfying the above constraints, print "NO" (without quotes) in one line.

Otherwise, print "YES" (without quotes) in the first line. Then in the second line print $a + b + c + d$ integers, separated by spaces — a beautiful sequence. There should be a numbers equal to 0, b numbers equal to 1, c numbers equal to 2 and d numbers equal to 3.

If there are multiple answers, you can print any of them.

Examples

input

```
2 2 2 1
```

output

```
YES
0 1 0 1 2 3 2
```

input

```
1 2 3 4
```

output

```
NO
```

input

```
2 2 2 3
```

output

```
NO
```

Note

In the first test, it is easy to see, that the sequence is beautiful because the difference between any two consecutive numbers is equal to 1. Also, there are exactly two numbers, equal to 0, 1, 2 and exactly one number, equal to 3.

It can be proved, that it is impossible to construct beautiful sequences in the second and third tests.

E. Beautiful Mirrors

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Creatnx has n mirrors, numbered from 1 to n. Every day, Creatnx asks exactly one mirror "Am I beautiful?". The i-th mirror will tell Creatnx that he is beautiful with probability $\frac{p_i}{100}$ for all $1 \le i \le n$.

Creatnx asks the mirrors one by one, starting from the 1-st mirror. Every day, if he asks i-th mirror, there are two possibilities:

- The i-th mirror tells Creatnx that he is beautiful. In this case, if $i = n$ Creatnx will stop and become happy, otherwise he will continue asking the $i+1$-th mirror next day;
- In the other case, Creatnx will feel upset. The next day, Creatnx will start asking from the 1-st mirror again.

You need to calculate the expected number of days until Creatnx becomes happy.

This number should be found by modulo 998244353. Formally, let $M = 998244353$. It can be shown that the answer can be expressed as an irreducible fraction $\frac{p}{q}$, where p and q are integers and $q \not\equiv 0 \pmod{M}$. Output the integer equal to $p \cdot q^{-1} \bmod M$. In other words, output such an integer x that $0 \le x < M$ and $x \cdot q \equiv p \pmod{M}$.

Input

The first line contains one integer n ($1 \le n \le 2 \cdot 10^5$) — the number of mirrors.

The second line contains n integers $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le 100$).

Output

Print the answer modulo 998244353 in a single line.

Examples

input

```
1
50
```

output

```
2
```

input

```
3
10 20 50
```

output

```
112
```

Note

In the first test, there is only one mirror and it tells, that Creatnx is beautiful with probability $\frac{1}{2}$. So, the expected number of days until Creatnx becomes happy is 2.

F. Beautiful Bracket Sequence (easy version)

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

This is the easy version of this problem. The only difference is the limit of n - the length of the input string. In this version, $1 \leq n \leq 2000$. The hard version of this challenge is not offered in the round for the second division.

Let's define a correct bracket sequence and its depth as follow:

- An empty string is a correct bracket sequence with depth 0.
- If "`s`" is a correct bracket sequence with depth d then "`(s)`" is a correct bracket sequence with depth $d+1$.
- If "`s`" and "`t`" are both correct bracket sequences then their concatenation "`st`" is a correct bracket sequence with depth equal to the maximum depth of s and t.

For a (not necessarily correct) bracket sequence s, we define its depth as the maximum depth of any **correct** bracket sequence induced by removing some characters from s (possibly zero). For example: the bracket sequence $s =$"`())(())`" has depth 2, because by removing the third character we obtain a correct bracket sequence "`()(())`" with depth 2.

Given a string a consists of only characters '`(`', '`)`' and '?'. Consider all (not necessarily correct) bracket sequences obtained by replacing all characters '?' in a by either '`(`' or '`)`'. Calculate the sum of all the depths of all these bracket sequences. As this number can be large, find it modulo 998244353.

Hacks in this problem in the first division can be done only if easy and hard versions of this problem was solved.

Input

The only line contains a non-empty string consist of only '`(`', '`)`' and '?'. The length of the string is at most 2000.

Output

Print the answer modulo 998244353 in a single line.

Examples

input

```
??
```

output

```
1
```

input

```
(?(?))
```

output

```
9
```

Note

In the first test case, we can obtain 4 bracket sequences by replacing all characters '?' with either '`(`' or '`)`':

- "`((`". Its depth is 0;
- "`))`". Its depth is 0;
- "`)(`". Its depth is 0;
- "`()`". Its depth is 1.

So, the answer is $1 = 0 + 0 + 0 + 1$.

In the second test case, we can obtain 4 bracket sequences by replacing all characters '?' with either '`(`' or '`)`':

- "`(((())`". Its depth is 2;
- "`()()))`". Its depth is 2;
- "`((()))`". Its depth is 3;
- "`()(())`". Its depth is 2.

So, the answer is $9 = 2 + 2 + 3 + 2$.