A. Luntik and Concerts

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

Luntik has decided to try singing. He has a one-minute songs, b two-minute songs and c three-minute songs. He wants to distribute all songs into two concerts such that every song should be included to exactly one concert.

He wants to make the absolute difference of durations of the concerts as small as possible. The duration of the concert is the sum of durations of all songs in that concert.

Please help Luntik and find the minimal possible difference in minutes between the concerts durations.

Input

The first line contains a single integer t $(1 \le t \le 1000)$ — the number of test cases.

Each test case consists of one line containing three integers a, b, c $(1 \le a, b, c \le 10^9)$ — the number of one-minute, two-minute and three-minute songs.

Output

For each test case print the minimal possible difference in minutes between the concerts durations.

Example

input

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

output

```
0
1
0
1
```

Note

In the first test case, Luntik can include a one-minute song and a two-minute song into the first concert, and a three-minute song into the second concert. Then the difference will be equal to 0.

In the second test case, Luntik can include two one-minute songs and a two-minute song and a three-minute song into the first concert, and two three-minute songs into the second concert. The duration of the first concert will be $1 + 1 + 2 + 3 = 7$, the duration of the second concert will be 6. The difference of them is $|7 - 6| = 1$.

B. Luntik and Subsequences

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

Luntik came out for a morning stroll and found an array a of length n. He calculated the sum s of the elements of the array ($s = \sum_{i=1}^{n} a_i$). Luntik calls a subsequence of the array a *nearly full* if the sum of the numbers in that subsequence is equal to $s - 1$.

Luntik really wants to know the number of *nearly full* subsequences of the array a. But he needs to come home so he asks you to solve that problem!

A sequence x is a subsequence of a sequence y if x can be obtained from y by deletion of several (possibly, zero or all) elements.

Input

The first line contains a single integer t $(1 \le t \le 1000)$ — the number of test cases. The next $2 \cdot t$ lines contain descriptions of test cases. The description of each test case consists of two lines.

The first line of each test case contains a single integer n $(1 \le n \le 60)$ — the length of the array.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(0 \le a_i \le 10^9)$ — the elements of the array a.

Output

For each test case print the number of *nearly full* subsequences of the array.

Example

input

```
5
5
1 2 3 4 5
2
1000 1000
2
1 0
5
3 0 2 1 1
5
2 1 0 3 0
```

output

```
1
0
2
4
4
```

Note

In the first test case, $s = 1 + 2 + 3 + 4 + 5 = 15$, only $(2, 3, 4, 5)$ is a nearly full subsequence among all subsequences, the sum in it is equal to $2 + 3 + 4 + 5 = 14 = 15 - 1$.

In the second test case, there are no nearly full subsequences.

In the third test case, $s = 1 + 0 = 1$, the nearly full subsequences are (0) and $()$ (the sum of an empty subsequence is 0).

C. Grandma Capa Knits a Scarf

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

Grandma Capa has decided to knit a scarf and asked Grandpa Sher to make a pattern for it, a pattern is a string consisting of lowercase English letters. Grandpa Sher wrote a string s of length n.

Grandma Capa wants to knit a beautiful scarf, and in her opinion, a beautiful scarf can only be knit from a string that is a palindrome. She wants to change the pattern written by Grandpa Sher, but to avoid offending him, she will choose one lowercase English letter and erase some (at her choice, possibly none or all) occurrences of that letter in string s.

She also wants to minimize the number of erased symbols from the pattern. Please help her and find the minimum number of symbols she can erase to make string s a palindrome, or tell her that it's impossible. Notice that she can only erase symbols equal to the **one** letter she chose.

A string is a palindrome if it is the same from the left to the right and from the right to the left. For example, the strings `'kek'`, `'abacaba'`, `'r'` and `'papicipap'` are palindromes, while the strings `'abb'` and `'iq'` are not.

Input

The first line contains a single integer t $(1 \le t \le 100)$ — the number of test cases. The next $2 \cdot t$ lines contain the description of test cases. The description of each test case consists of two lines.

The first line of each test case contains a single integer n $(1 \le n \le 10^5)$ — the length of the string.

The second line of each test case contains the string s consisting of n lowercase English letters.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case print the minimum number of erased symbols required to make the string a palindrome, if it is possible, and -1, if it is impossible.

Example

input

```
5
8
abcaacab
6
xyzxyz
4
abba
8
rprarlap
10
khyyhhyhky
```

output

```
2
-1
0
3
2
```

Note

In the first test case, you can choose a letter `'a'` and erase its first and last occurrences, you will get a string `'bcaacb'`, which is a palindrome. You can also choose a letter `'b'` and erase all its occurrences, you will get a string `'acaaca'`, which is a palindrome as well.

In the second test case, it can be shown that it is impossible to choose a letter and erase some of its occurrences to get a palindrome.

In the third test case, you don't have to erase any symbols because the string is already a palindrome.

D. Vupsen, Pupsen and 0

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

Vupsen and Pupsen were gifted an integer array. Since Vupsen doesn't like the number 0, he threw away all numbers equal to 0 from the array. As a result, he got an array a of length n.

Pupsen, on the contrary, likes the number 0 and he got upset when he saw the array without zeroes. To cheer Pupsen up, Vupsen decided to come up with another array b of length n such that $\sum_{i=1}^{n} a_i \cdot b_i = 0$. Since Vupsen doesn't like number 0, **the array b must not contain numbers equal to 0**. Also, the numbers in that array must not be huge, so **the sum of their absolute values cannot exceed 10^9**. Please help Vupsen to find any such array b!

Input

The first line contains a single integer t ($1 \le t \le 100$) — the number of test cases. The next $2 \cdot t$ lines contain the description of test cases. The description of each test case consists of two lines.

The first line of each test case contains a single integer n ($2 \le n \le 10^5$) — the length of the array.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($-10^4 \le a_i \le 10^4$, $a_i \ne 0$) — the elements of the array a.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case print n integers $b_1, b_2, \ldots, b_n$ — elements of the array b ($|b_1| + |b_2| + \ldots + |b_n| \le 10^9$, $b_i \ne 0$, $\sum_{i=1}^{n} a_i \cdot b_i = 0$).

It can be shown that the answer always exists.

Example

input

```
3
2
5 5
5
5 -2 10 -9 4
7
1 2 3 4 5 6 7
```

output

```
1 -1
-1 5 1 -1 -1
-10 2 2 -3 5 -1 -1
```

Note

In the first test case, $5 \cdot 1 + 5 \cdot (-1) = 5 - 5 = 0$. You could also print 3 -3, for example, since $5 \cdot 3 + 5 \cdot (-3) = 15 - 15 = 0$

In the second test case, $5 \cdot (-1) + (-2) \cdot 5 + 10 \cdot 1 + (-9) \cdot (-1) + 4 \cdot (-1) = -5 - 10 + 10 + 9 - 4 = 0$.

E. Pchelyonok and Segments

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Pchelyonok decided to give Mila a gift. Pchelenok has already bought an array a of length n, but gifting an array is too common. Instead of that, he decided to gift Mila the segments of that array!

Pchelyonok wants his gift to be beautiful, so he decided to choose k non-overlapping segments of the array $[l_1, r_1]$, $[l_2, r_2]$, $\ldots$

$[l_k, r_k]$ such that:

- the length of the first segment $[l_1, r_1]$ is k, the length of the second segment $[l_2, r_2]$ is $k-1$, $\dots$, the length of the k-th segment $[l_k, r_k]$ is 1
- for each $i < j$, the i-th segment occurs in the array earlier than the j-th (i.e. $r_i < l_j$)
- the sums in these segments are strictly increasing (i.e. let $sum(l \dots r) = \sum\limits_{i=l}^{r} a_i$ — the sum of numbers in the segment $[l, r]$ of the array, then $sum(l_1 \dots r_1) < sum(l_2 \dots r_2) < \dots < sum(l_k \dots r_k)$).

Pchelenok also wants his gift to be as beautiful as possible, so he asks you to find the maximal value of k such that he can give Mila a gift!

Input

The first line contains a single integer t $(1 \le t \le 100)$ — the number of test cases. The next $2 \cdot t$ lines contain the descriptions of test cases. The description of each test case consists of two lines.

The first line of each test case contains a single integer n $(1 \le n \le 10^5)$ — the length of the array.

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

It is guaranteed that the sum of n over all test cases does not exceed 10^5.

Output

For each test case, print the maximum possible value of k.

Example

input

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

output

```
1
1
2
3
1
```

F1. Korney Korneevich and XOR (easy version)

time limit per test: 1.5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

This is an easier version of the problem with smaller constraints.

Korney Korneevich dag up an array a of length n. Korney Korneevich has recently read about the operation bitwise XOR, so he wished to experiment with it. For this purpose, he decided to find all integers $x \ge 0$ such that there exists an **increasing** subsequence of the array a, in which the bitwise XOR of numbers is equal to x.

It didn't take a long time for Korney Korneevich to find all such x, and he wants to check his result. That's why he asked you to solve this problem!

A sequence s is a subsequence of a sequence b if s can be obtained from b by deletion of several (possibly, zero or all) elements.

A sequence $s_1, s_2, \dots, s_m$ is called increasing if $s_1 < s_2 < \dots < s_m$.

Input

The first line contains a single integer n $(1 \le n \le 10^5)$.

The second line contains n integers $a_1, a_2, \dots, a_n$ $(0 \le a_i \le 500)$ — the elements of the array a.

Output

In the first line print a single integer k — the number of found x values.

In the second line print k integers in **increasing** order $x_1, x_2, \dots x_k$ $(0 \le x_1 < \dots < x_k)$ — found x values.

Examples

input

```
4
4 2 2 4
```

output

```
4
0 2 4 6
```

input

```
8
1 0 1 7 12 5 3 2
```

output

```
12
0 1 2 3 4 5 6 7 10 11 12 13
```

Note

In the first test case:

- To get value $x = 0$ it is possible to choose and empty subsequence
- To get value $x = 2$ it is possible to choose a subsequence $[2]$
- To get value $x = 4$ it is possible to choose a subsequence $[4]$
- To get value $x = 6$ it is possible to choose a subsequence $[2, 4]$

F2. Korney Korneevich and XOR (hard version)

time limit per test: 1.5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

This is a harder version of the problem with bigger constraints.

Korney Korneevich dag up an array a of length n. Korney Korneevich has recently read about the operation bitwise XOR, so he wished to experiment with it. For this purpose, he decided to find all integers $x \geq 0$ such that there exists an **increasing** subsequence of the array a, in which the bitwise XOR of numbers is equal to x.

It didn't take a long time for Korney Korneevich to find all such x, and he wants to check his result. That's why he asked you to solve this problem!

A sequence s is a subsequence of a sequence b if s can be obtained from b by deletion of several (possibly, zero or all) elements.

A sequence $s_1, s_2, \ldots, s_m$ is called increasing if $s_1 < s_2 < \ldots < s_m$.

Input

The first line contains a single integer n ($1 \leq n \leq 10^6$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \leq a_i \leq 5000$) — the elements of the array a.

Output

In the first line print a single integer k — the number of found x values.

In the second line print k integers in **increasing** order $x_1, x_2, \ldots x_k$ ($0 \leq x_1 < \ldots < x_k$) — found x values.

Examples

input

```
4
4 2 2 4
```

output

```
4
0 2 4 6
```

input

```
8
1 0 1 7 12 5 3 2
```

output

```
12
0 1 2 3 4 5 6 7 10 11 12 13
```

Note

In the first test case:

- To get value $x = 0$ it is possible to choose and empty subsequence
- To get value $x = 2$ it is possible to choose a subsequence $[2]$
- To get value $x = 4$ it is possible to choose a subsequence $[4]$
- To get value $x = 6$ it is possible to choose a subsequence $[2, 4]$

G. Kuzya and Homework

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Kuzya started going to school. He was given math homework in which he was given an array a of length n and an array of symbols b of length n, consisting of symbols '*' and '/'.

Let's denote a *path of calculations* for a segment $[l; r]$ ($1 \le l \le r \le n$) in the following way:

- Let $x = 1$ initially. For every i from l to r we will consequently do the following: if $b_i =$ '*', $x = x * a_i$, and if $b_i =$ '/', then $x = \frac{x}{a_i}$. Let's call *a path of calculations* for the segment $[l; r]$ a list of all x that we got during the calculations (the number of them is exactly $r - l + 1$).

For example, let $a = [7, 12, 3, 5, 4, 10, 9]$, $b = [/, *, /, /, /, *, *]$, $l = 2$, $r = 6$, then the path of calculations for that segment is $[12, 4, 0.8, 0.2, 2]$.

Let's call a segment $[l; r]$ *simple* if the path of calculations for it contains **only integer numbers**.

Kuzya needs to find the number of simple segments $[l; r]$ ($1 \le l \le r \le n$). Since he obviously has no time and no interest to do the calculations for each option, he asked you to write a program to get to find that number!

Input

The first line contains a single integer n ($2 \le n \le 10^6$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^6$).

The third line contains n symbols without spaces between them — the array $b_1, b_2 \ldots b_n$ ($b_i =$ '/' or $b_i =$ '*' for every $1 \le i \le n$).

Output

Print a single integer — the number of simple segments $[l; r]$.

Examples

input

```
3
1 2 3
*/*
```

output

```
2
```

input

```
7
6 4 10 1 2 15 1
*/*/*//
```

output

```
8
```