Codeforces Round #587 (Div. 3)

A. Prefixes

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

Nikolay got a string s of **even** length n, which consists only of lowercase Latin letters 'a' and 'b'. Its positions are numbered from 1 to n.

He wants to modify his string so that every its prefix of **even** length has an equal amount of letters 'a' and 'b'. To achieve that, Nikolay can perform the following operation arbitrary number of times (possibly, zero): choose some position in his string and replace the letter on this position with the other letter (i.e. replace 'a' with 'b' or replace 'b' with 'a'). Nikolay can use no letters except 'a' and 'b'.

The prefix of string s of length l ($1 \le l \le n$) is a string $s[1..l]$.

For example, for the string $s=$"abba" there are two prefixes of the even length. The first is $s[1\dots2]=$"ab" and the second $s[1\dots4]=$"abba". Both of them have the same number of 'a' and 'b'.

Your task is to calculate the minimum number of operations Nikolay has to perform with the string s to modify it so that every its prefix of **even** length has an equal amount of letters 'a' and 'b'.

Input

The first line of the input contains one **even** integer n $(2 \le n \le 2 \cdot 10^5)$ — the length of string s.

The second line of the input contains the string s of length n, which consists only of lowercase Latin letters 'a' and 'b'.

Output

In the first line print the minimum number of operations Nikolay has to perform with the string s to modify it so that every its prefix of **even** length has an equal amount of letters 'a' and 'b'.

In the second line print the string Nikolay obtains after applying all the operations. If there are multiple answers, you can print any of them.

Examples

input
```
4
bbbb
```
output
```
2
abba
```

input
```
6
ababab
```
output
```
0
ababab
```

input
```
2
aa
```
output
```
1
ba
```

Note

In the first example Nikolay has to perform two operations. For example, he can replace the first 'b' with 'a' and the last 'b' with 'a'.

In the second example Nikolay doesn't need to do anything because each prefix of an even length of the initial string already contains an equal amount of letters 'a' and 'b'.

B. Shooting

time limit per test: 1 second

memory limit per test: 256 megabytes
input: standard input
output: standard output

Recently Vasya decided to improve his pistol shooting skills. Today his coach offered him the following exercise. He placed n cans in a row on a table. Cans are numbered from left to right from 1 to n. Vasya has to knock down each can exactly once to finish the exercise. He is allowed to choose **the order** in which he will knock the cans down.

Vasya knows that the *durability* of the i-th can is a_i. It means that if Vasya has already knocked x cans down and is now about to start shooting the i-th one, he will need $(a_i \cdot x + 1)$ shots to knock it down. You can assume that if Vasya starts shooting the i-th can, he will be shooting it until he knocks it down.

Your task is to choose such an order of shooting so that the number of shots required to knock each of the n given cans down exactly once is minimum possible.

Input

The first line of the input contains one integer n $(2 \le n \le 1\,000)$ — the number of cans.

The second line of the input contains the sequence $a_1, a_2, \dots, a_n$ $(1 \le a_i \le 1\,000)$, where a_i is the durability of the i-th can.

Output

In the first line print the minimum number of shots required to knock each of the n given cans down exactly once.

In the second line print the sequence consisting of n **distinct** integers from 1 to n — the order of indices of cans that minimizes the number of shots required. If there are several answers, you can print any of them.

Examples

input
```
3
20 10 20
```
output
```
43
1 3 2
```

input
```
4
10 10 10 10
```
output
```
64
2 1 4 3
```

input
```
6
5 4 5 4 4 5
```
output
```
69
6 1 3 5 2 4
```

input
```
2
1 4
```
output
```
3
2 1
```

Note

In the first example Vasya can start shooting from the first can. He knocks it down with the first shot because he haven't knocked any other cans down before. After that he has to shoot the third can. To knock it down he shoots $20 \cdot 1 + 1 = 21$ times. After that only second can remains. To knock it down Vasya shoots $10 \cdot 2 + 1 = 21$ times. So the total number of shots is $1 + 21 + 21 = 43$.

In the second example the order of shooting does not matter because all cans have the same durability.

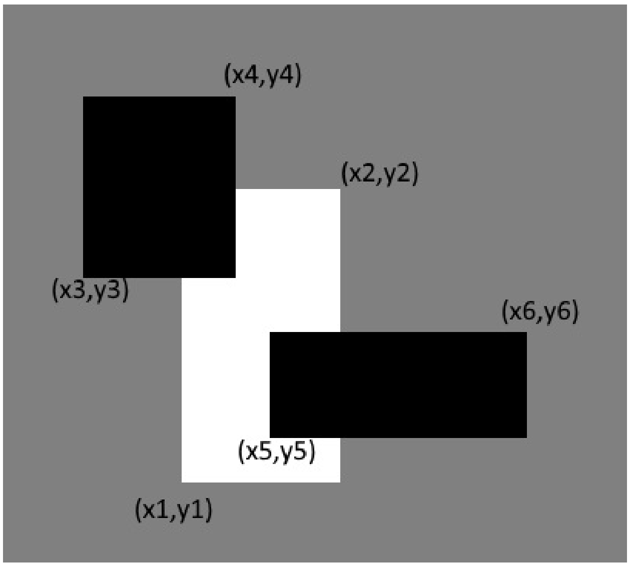
C. White Sheet

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

There is a white sheet of paper lying on a rectangle table. The sheet is a rectangle with its sides parallel to the sides of the table. If you will take a look from above and assume that the bottom left corner of the table has coordinates $(0, 0)$, and coordinate axes are left and bottom sides of the table, then the bottom left corner of the white sheet has coordinates (x_1, y_1), and the top right —

(x_2, y_2).

After that two black sheets of paper are placed on the table. Sides of both black sheets are also parallel to the sides of the table. Coordinates of the bottom left corner of the first black sheet are (x_3, y_3), and the top right — (x_4, y_4). Coordinates of the bottom left corner of the second black sheet are (x_5, y_5), and the top right — (x_6, y_6).



Example of three rectangles.

Determine if some part of the white sheet can be seen from the above after the two black sheets are placed. The part of the white sheet can be seen if there is at least one point lying **not strictly inside** the white sheet and **strictly outside** of both black sheets.

Input

The first line of the input contains four integers x_1, y_1, x_2, y_2 $(0 \le x_1 < x_2 \le 10^6, 0 \le y_1 < y_2 \le 10^6)$ — coordinates of the bottom left and the top right corners of the white sheet.

The second line of the input contains four integers x_3, y_3, x_4, y_4 $(0 \le x_3 < x_4 \le 10^6, 0 \le y_3 < y_4 \le 10^6)$ — coordinates of the bottom left and the top right corners of the first black sheet.

The third line of the input contains four integers x_5, y_5, x_6, y_6 $(0 \le x_5 < x_6 \le 10^6, 0 \le y_5 < y_6 \le 10^6)$ — coordinates of the bottom left and the top right corners of the second black sheet.

The sides of each sheet of paper are parallel (perpendicular) to the coordinate axes.

Output

If some part of the white sheet can be seen from the above after the two black sheets are placed, print "YES" (without quotes). Otherwise print "NO".

Examples

input

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

output

```
NO
```

input

```
3 3 7 5
0 0 4 6
0 0 7 4
```

output

```
YES
```

input

```
5 2 10 5
3 1 7 6
8 1 11 7
```

output

```
YES
```

input

```
0 0 1000000 1000000
0 0 499999 1000000
500000 0 1000000 1000000
```

output

```
YES
```

Note

In the first example the white sheet is fully covered by black sheets.

In the second example the part of the white sheet can be seen after two black sheets are placed. For example, the point $(6.5, 4.5)$ lies not strictly inside the white sheet and lies strictly outside of both black sheets.

D. Swords

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

There were n types of swords in the theater basement which had been used during the plays. Moreover there were **exactly** x swords of each type. y people have broken into the theater basement and each of them has taken exactly z swords of some **single type**. Note that different people might have taken different types of swords. Note that the values x, y and z are unknown for you.

The next morning the director of the theater discovers the loss. He counts all swords — exactly a_i swords of the i-th type are left untouched.

The director has no clue about the initial number of swords of each type in the basement, the number of people who have broken into the basement and how many swords each of them have taken.

For example, if $n = 3$, $a = [3, 12, 6]$ then one of the possible situations is $x = 12$, $y = 5$ and $z = 3$. Then the first three people took swords of the first type and the other two people took swords of the third type. Note that you don't know values x, y and z beforehand but know values of n and a.

Thus he seeks for your help. Determine the **minimum** number of people y, which could have broken into the theater basement, and the number of swords z each of them has taken.

Input

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

The second line of the input contains the sequence $a_1, a_2, \dots, a_n$ $(0 \le a_i \le 10^9)$, where a_i equals to the number of swords of the i-th type, which have remained in the basement after the theft. It is guaranteed that there exists at least one such pair of indices (j, k) that $a_j \ne a_k$.

Output

Print two integers y and z — the minimum number of people which could have broken into the basement and the number of swords each of them has taken.

Examples

input

```
3
3 12 6
```

output

```
5 3
```

input

```
2
2 9
```

output

```
1 7
```

input

```
7
2 1000000000 4 6 8 4 2
```

output

```
2999999987 2
```

input

```
6
13 52 0 13 26 52
```

output

```
12 13
```

Note

In the first example the minimum value of y equals to 5, i.e. the minimum number of people who could have broken into the basement, is 5. Each of them has taken 3 swords: three of them have taken 3 swords of the first type, and two others have taken 3

swords of the third type.

In the second example the minimum value of y is 1, i.e. the minimum number of people who could have broken into the basement, equals to 1. He has taken 7 swords of the first type.

E1. Numerical Sequence (easy version)

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

The only difference between the easy and the hard versions is the maximum value of k.

You are given an *infinite* sequence of form "112123123412345..." which consist of blocks of all consecutive positive integers written one after another. The first block consists of all numbers from 1 to 1, the second one — from 1 to 2, the third one — from 1 to 3, $\dots$, the i-th block consists of all numbers from 1 to i.

So the first 56 elements of the sequence are "11212312341234512345612345671234567812345678912345678910". Elements of the sequence are numbered from one. For example, the 1-st element of the sequence is 1, the 3-rd element of the sequence is 2, the 20-th element of the sequence is 5, the 38-th element is 2, the 56-th element of the sequence is 0.

Your task is to answer q independent queries. In the i-th query you are given one integer k_i. Calculate the digit at the position k_i of the sequence.

Input

The first line of the input contains one integer q ($1 \le q \le 500$) — the number of queries.

The i-th of the following q lines contains one integer k_i $(1 \le k_i \le 10^9)$ — the description of the corresponding query.

Output

Print q lines. In the i-th line print one digit x_i $(0 \le x_i \le 9)$ — the answer to the query i, i.e. x_i should be equal to the element at the position k_i of the sequence.

Examples

input

```
5
1
3
20
38
56
```

output

```
1
2
5
2
0
```

input

```
4
2132
506
999999999
1000000000
```

output

```
8
2
9
8
```

Note

Answers on queries from the first example are described in the problem statement.

E2. Numerical Sequence (hard version)

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

The only difference between the easy and the hard versions is the maximum value of k.

You are given an *infinite* sequence of form "112123123412345..." which consist of blocks of all consecutive positive integers written one after another. The first block consists of all numbers from 1 to 1, the second one — from 1 to 2, the third one — from 1 to 3, $\dots$, the i-th block consists of all numbers from 1 to i.

So the first 56 elements of the sequence are "11212312341234512345612345671234567812345678912345678910". Elements of the sequence are numbered from one. For example, the 1-st element of the sequence is 1, the 3-rd element of the sequence is 2, the 20-th element of the sequence is 5, the 38-th element is 2, the 56-th element of the sequence is 0.

Your task is to answer q independent queries. In the i-th query you are given one integer k_i. Calculate the digit at the position k_i of the sequence.

Input

The first line of the input contains one integer q ($1 \le q \le 500$) — the number of queries.

The i-th of the following q lines contains one integer k_i $(1 \le k_i \le 10^{18})$ — the description of the corresponding query.

Output

Print q lines. In the i-th line print one digit x_i $(0 \le x_i \le 9)$ — the answer to the query i, i.e. x_i should be equal to the element at the position k_i of the sequence.

Examples

input

```
5
1
3
20
38
56
```

output

```
1
2
5
2
0
```

input

```
4
2132
506
999999999999999999
1000000000000000000
```

output

```
8
2
4
1
```

Note

Answers on queries from the first example are described in the problem statement.

F. Wi-Fi

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

You work as a system administrator in a dormitory, which has n rooms one after another along a straight hallway. Rooms are numbered from 1 to n.

You have to connect all n rooms to the Internet.

You can connect each room to the Internet directly, the cost of such connection for the i-th room is i coins.

Some rooms also have a spot for a router. The cost of placing a router in the i-th room is also i coins. You cannot place a router in a room which does not have a spot for it. When you place a router in the room i, you connect all rooms with the numbers from $max(1,\ i-k)$ to $min(n,\ i+k)$ inclusive to the Internet, where k is the range of router. The value of k is the same for all routers.

Calculate the minimum total cost of connecting all n rooms to the Internet. You can assume that the number of rooms which have a spot for a router is not greater than the number of routers you have.

Input

The first line of the input contains two integers n and k ($1 \le n, k \le 2 \cdot 10^5$) — the number of rooms and the range of each router.

The second line of the input contains one string s of length n, consisting only of zeros and ones. If the i-th character of the string equals to '1' then there is a spot for a router in the i-th room. If the i-th character of the string equals to '0' then you cannot place a router in the i-th room.

Output

Print one integer — the minimum total cost of connecting all n rooms to the Internet.

Examples

input

```
5 2
00100
```

output

```
3
```

input

```
6 1
000000
```

output

```
21
```

input

```
4 1
0011
```

output

```
4
```

input

```
12 6
000010000100
```

output

```
15
```

Note

In the first example it is enough to place the router in the room 3, then all rooms will be connected to the Internet. The total cost of connection is 3.

In the second example you can place routers nowhere, so you need to connect all rooms directly. Thus, the total cost of connection of all rooms is $1 + 2 + 3 + 4 + 5 + 6 = 21$.

In the third example you need to connect the room 1 directly and place the router in the room 3. Thus, the total cost of connection of all rooms is $1 + 3 = 4$.

In the fourth example you need to place routers in rooms 5 and 10. Then all rooms will be connected to the Internet. The total cost of connection is $5 + 10 = 15$.