A. Co-prime Array

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

You are given an array of n elements, you must make it a co-prime array in as few moves as possible.

In each move you can insert any positive integral number you want not greater than 10^9 in any place in the array.

An array is co-prime if any two adjacent numbers of it are co-prime.

In the number theory, two integers a and b are said to be co-prime if the only positive integer that divides both of them is 1.

Input

The first line contains integer n ($1 \le n \le 1000$) — the number of elements in the given array.

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

Output

Print integer k on the first line — the least number of elements needed to add to the array a to make it co-prime.

The second line should contain $n + k$ integers a_j — the elements of the array a after adding k elements to it. Note that the new array should be co-prime, so any two adjacent values should be co-prime. Also the new array should be got from the original array a by adding k elements to it.

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

Example

input

```
3
2 7 28
```

output

```
1
2 7 9 28
```

B. Seating On Bus

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

Consider $2n$ rows of the seats in a bus. n rows of the seats on the left and n rows of the seats on the right. Each row can be filled by two people. So the total capacity of the bus is $4n$.

Consider that m ($m \le 4n$) people occupy the seats in the bus. The passengers entering the bus are numbered from 1 to m (in the order of their entering the bus). The pattern of the seat occupation is as below:

1-st row left window seat, 1-st row right window seat, 2-nd row left window seat, 2-nd row right window seat, ... , n-th row left window seat, n-th row right window seat.

After occupying all the window seats (for $m > 2n$) the non-window seats are occupied:

1-st row left non-window seat, 1-st row right non-window seat, ... , n-th row left non-window seat, n-th row right non-window seat.

All the passengers go to a single final destination. In the final destination, the passengers get off in the given order.

1-st row left non-window seat, 1-st row left window seat, 1-st row right non-window seat, 1-st row right window seat, ... , n-th row left non-window seat, n-th row left window seat, n-th row right non-window seat, n-th row right window seat.

The seating for $n = 9$ and $m = 36$.

You are given the values n and m. Output m numbers from 1 to m, the order in which the passengers will get off the bus.

Input

The only line contains two integers, n and m ($1 \le n \le 100, 1 \le m \le 4n$) — the number of pairs of rows and the number of passengers.

Output

Print m distinct integers from 1 to m — the order in which the passengers will get off the bus.

Examples

input
```
2 7
```
output
```
5 1 6 2 7 3 4
```

input
```
9 36
```
output
```
19 1 20 2 21 3 22 4 23 5 24 6 25 7 26 8 27 9 28 10 29 11 30 12 31 13 32 14 33 15 34 16 35 17 36 18
```

C. Hard Process

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

You are given an array a with n elements. Each element of a is either 0 or 1.

Let's denote the length of the longest subsegment of consecutive elements in a, consisting of only numbers one, as $f(a)$. You can change no more than k zeroes to ones to maximize $f(a)$.

Input

The first line contains two integers n and k ($1 \le n \le 3 \cdot 10^5, 0 \le k \le n$) — the number of elements in a and the parameter k.

The second line contains n integers a_i ($0 \le a_i \le 1$) — the elements of a.

Output

On the first line print a non-negative integer z — the maximal value of $f(a)$ after no more than k changes of zeroes to ones.

On the second line print n integers a_j — the elements of the array a after the changes.

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

Examples

input

```
7 1
1 0 0 1 1 0 1
```

output

```
4
1 0 0 1 1 1 1
```

input

```
10 2
1 0 0 1 0 1 0 1 0 1
```

output

```
5
1 0 0 1 1 1 1 1 0 1
```

D. Number of Parallelograms

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given n points on a plane. All the points are distinct and no three of them lie on the same line. Find the number of parallelograms with the vertices at the given points.

Input

The first line of the input contains integer n ($1 \le n \le 2000$) — the number of points.

Each of the next n lines contains two integers (x_i, y_i) ($0 \le x_i, y_i \le 10^9$) — the coordinates of the i-th point.

Output

Print the only integer c — the number of parallelograms with the vertices at the given points.

Example

input

```
4
0 1
1 0
1 1
2 0
```

output

```
1
```

E. Different Subsets For All Tuples

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

For a sequence a of n integers between 1 and m, inclusive, denote $f(a)$ as the number of distinct subsequences of a (including the empty subsequence).

You are given two positive integers n and m. Let S be the set of all sequences of length n consisting of numbers from 1 to m. Compute the sum $f(a)$ over all a in S modulo $10^9 + 7$.

Input

The only line contains two integers n and m ($1 \le n, m \le 10^6$) — the number of elements in arrays and the upper bound for elements.

Output

Print the only integer c — the desired sum modulo $10^9 + 7$.

Examples

input
```
1 3
```
output
```
6
```

input
```
2 2
```
output
```
14
```

input
```
3 3
```
output
```
174
```

F. Bear and Bowling 4

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

Limak is an old brown bear. He often goes bowling with his friends. Today he feels really good and tries to beat his own record!

For rolling a ball one gets a score — an integer (maybe negative) number of points. Score for the i-th roll is multiplied by i and scores are summed up. So, for k rolls with scores $s_1, s_2, \ldots, s_k$, the total score is $\sum_{i=1}^{k} i \cdot s_i$. The total score is 0 if there were no rolls.

Limak made n rolls and got score a_i for the i-th of them. He wants to maximize his total score and he came up with an interesting idea. He can say that some first rolls were only a warm-up, and that he wasn't focused during the last rolls. More formally, he can cancel any prefix and any suffix of the sequence $a_1, a_2, \ldots, a_n$. It is allowed to cancel all rolls, or to cancel none of them.

The total score is calculated as if there were only non-canceled rolls. So, the first non-canceled roll has score multiplied by 1, the second one has score multiplied by 2, and so on, till the last non-canceled roll.

What maximum total score can Limak get?

Input

The first line contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the total number of rolls made by Limak.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($|a_i| \le 10^7$) — scores for Limak's rolls.

Output

Print the maximum possible total score after cancelling rolls.

Examples

input

```
6
5 -1000 1 -3 7 -8
```

output

```
16
```

input

```
5
1000 1000 1001 1000 1000
```

output

```
15003
```

input

```
3
-60 -70 -80
```

output

```
0
```

Note

In the first sample test, Limak should cancel the first two rolls, and one last roll. He will be left with rolls $1, -3, 7$ what gives him the total score $1 \cdot 1 + 2 \cdot (-3) + 3 \cdot 7 = 1 - 6 + 21 = 16$.