A. A Twisty Movement

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

A dragon symbolizes wisdom, power and wealth. On Lunar New Year's Day, people model a dragon with bamboo strips and clothes, raise them with rods, and hold the rods high and low to resemble a flying dragon.

A performer holding the rod low is represented by a 1, while one holding it high is represented by a 2. Thus, the line of performers can be represented by a sequence $a_1, a_2, \ldots, a_n$.

Little Tommy is among them. He would like to choose an interval $[l, r]$ $(1 \le l \le r \le n)$, then reverse $a_l, a_{l+1}, \ldots, a_r$ so that the length of the longest non-decreasing subsequence of the new sequence is maximum.

A non-decreasing subsequence is a sequence of indices $p_1, p_2, \ldots, p_k$, such that $p_1 < p_2 < \ldots < p_k$ and $a_{p_1} \le a_{p_2} \le \ldots \le a_{p_k}$. The length of the subsequence is k.

Input

The first line contains an integer n $(1 \le n \le 2000)$, denoting the length of the original sequence.

The second line contains n space-separated integers, describing the original sequence $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 2, i = 1, 2, \ldots, n)$.

Output

Print a single integer, which means the maximum possible length of the longest non-decreasing subsequence of the new sequence.

Examples

input

```
4
1 2 1 2
```

output

```
4
```

input

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

output

```
9
```

Note

In the first example, after reversing $[2, 3]$, the array will become $[1, 1, 2, 2]$, where the length of the longest non-decreasing subsequence is 4.

In the second example, after reversing $[3, 7]$, the array will become $[1, 1, 1, 1, 2, 2, 2, 2, 2, 1]$, where the length of the longest non-decreasing subsequence is 9.

B. A Determined Cleanup

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

In order to put away old things and welcome a fresh new year, a thorough cleaning of the house is a must.

Little Tommy finds an old polynomial and cleaned it up by taking it modulo another. But now he regrets doing this...

Given two integers p and k, find a polynomial $f(x)$ with non-negative integer coefficients strictly less than k, whose remainder is p when divided by $(x + k)$. That is, $f(x) = q(x) \cdot (x + k) + p$, where $q(x)$ is a polynomial (not necessarily with integer coefficients).

Input

The only line of input contains two space-separated integers p and k $(1 \le p \le 10^{18}, 2 \le k \le 2\,000)$.

Output

If the polynomial does not exist, print a single integer `-1`, or output two lines otherwise.

In the first line print a non-negative integer d — the number of coefficients in the polynomial.

In the second line print d space-separated integers $a_0, a_1, \ldots, a_{d-1}$, describing a polynomial $f(x) = \sum_{i=0}^{d-1} a_i \cdot x^i$ fulfilling the given requirements. Your output should satisfy $0 \le a_i < k$ for all $0 \le i \le d - 1$, and $a_{d-1} \ne 0$.

If there are many possible solutions, print any of them.

Examples

input

```
46 2
```

output

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

input

```
2018 214
```

output

```
3
92 205 1
```

Note

In the first example, $f(x) = x^6 + x^5 + x^4 + x = (x^5 - x^4 + 3x^3 - 6x^2 + 12x - 23) \cdot (x + 2) + 46$.

In the second example, $f(x) = x^2 + 205x + 92 = (x - 9) \cdot (x + 214) + 2018$.

C. A Colourful Prospect

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

Firecrackers scare Nian the monster, but they're wayyyyy too noisy! Maybe fireworks make a nice complement.

Little Tommy is watching a firework show. As circular shapes spread across the sky, a splendid view unfolds on the night of Lunar New Year's eve.

A wonder strikes Tommy. How many regions are formed by the circles on the sky? We consider the sky as a flat plane. A region is a connected part of the plane with positive area, whose bound consists of parts of bounds of the circles and is a curve or several curves without self-intersections, and that does not contain any curve other than its boundaries. Note that exactly one of the regions extends infinitely.

Input

The first line of input contains one integer n ($1 \le n \le 3$), denoting the number of circles.

The following n lines each contains three space-separated integers x, y and r ($-10 \le x, y \le 10$, $1 \le r \le 10$), describing a circle whose center is (x, y) and the radius is r. No two circles have the same x, y and r at the same time.

Output

Print a single integer — the number of regions on the plane.

Examples

input

```
3
0 0 1
2 0 1
4 0 1
```

output

```
4
```

input

```
3
0 0 2
3 0 2
6 0 2
```

output

```
6
```

input

```
3
0 0 2
2 0 2
```

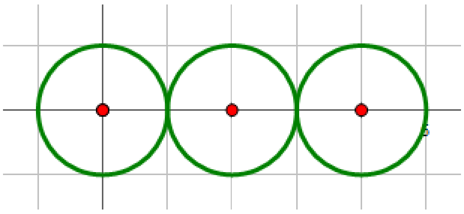
```
1 1 2
```

output

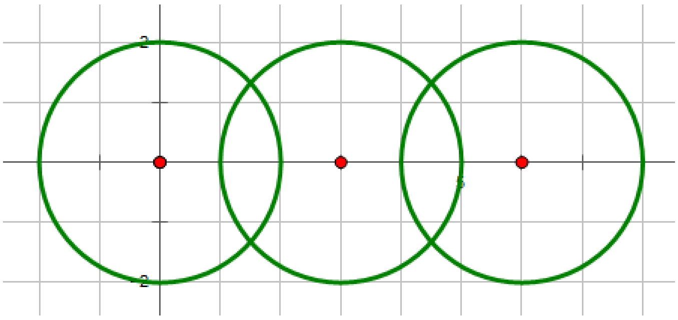
```
8
```

Note

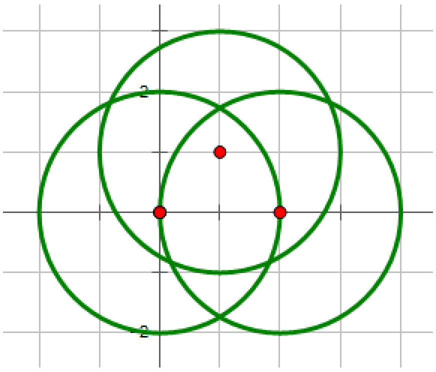
For the first example,



For the second example,



For the third example,



D. A Creative Cutout

time limit per test: 2 seconds

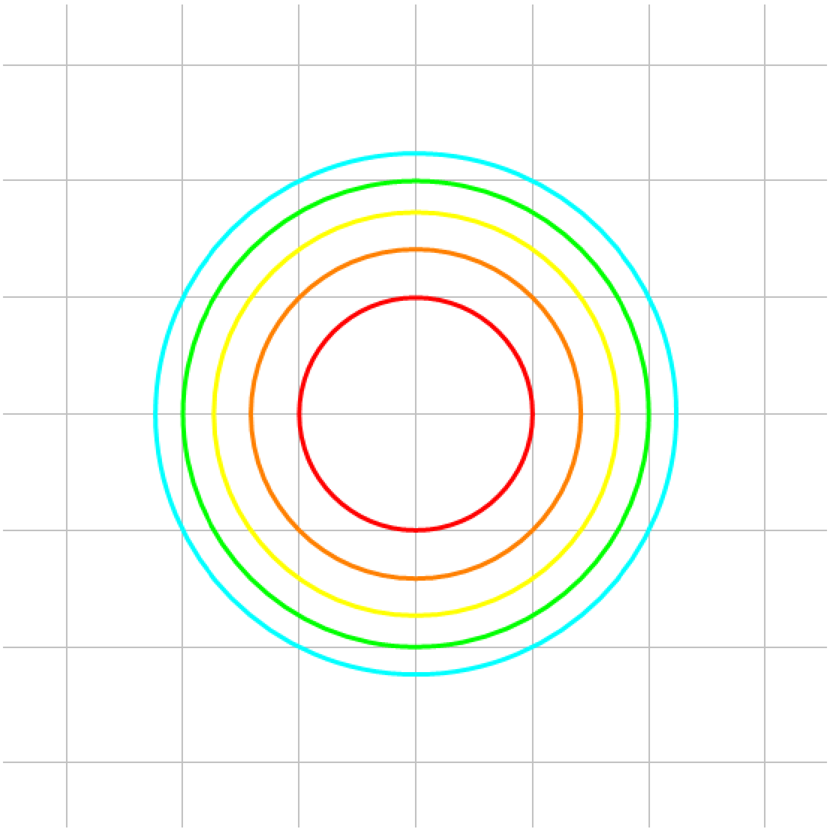
memory limit per test: 256 megabytes

input: standard input

output: standard output

Everything red frightens Nian the monster. So do red paper and... you, red on Codeforces, potential or real.

Big Banban has got a piece of paper with endless lattice points, where lattice points form squares with the same area. His most favorite closed shape is the circle because of its beauty and simplicity. Once he had obtained this piece of paper, he prepares it for paper-cutting.



He drew n concentric circles on it and numbered these circles from 1 to n such that the center of each circle is the same lattice point and the radius of the k-th circle is $\sqrt{k}$ times the length of a lattice edge.

Define the degree of beauty of a lattice point as the summation of the **indices** of circles such that this lattice point is inside them, or on their bounds. Banban wanted to ask you the total degree of beauty of all the lattice points, but changed his mind.

Defining the total degree of beauty of all the lattice points on a piece of paper with n circles as $f(n)$, you are asked to figure out $(\sum_{k=1}^{m} f(k)) \bmod (10^9+7)$.

Input

The first line contains one integer m $(1 \le m \le 10^{12})$.

Output

In the first line print one integer representing $(\sum_{k=1}^{m} f(k)) \bmod (10^9+7)$.

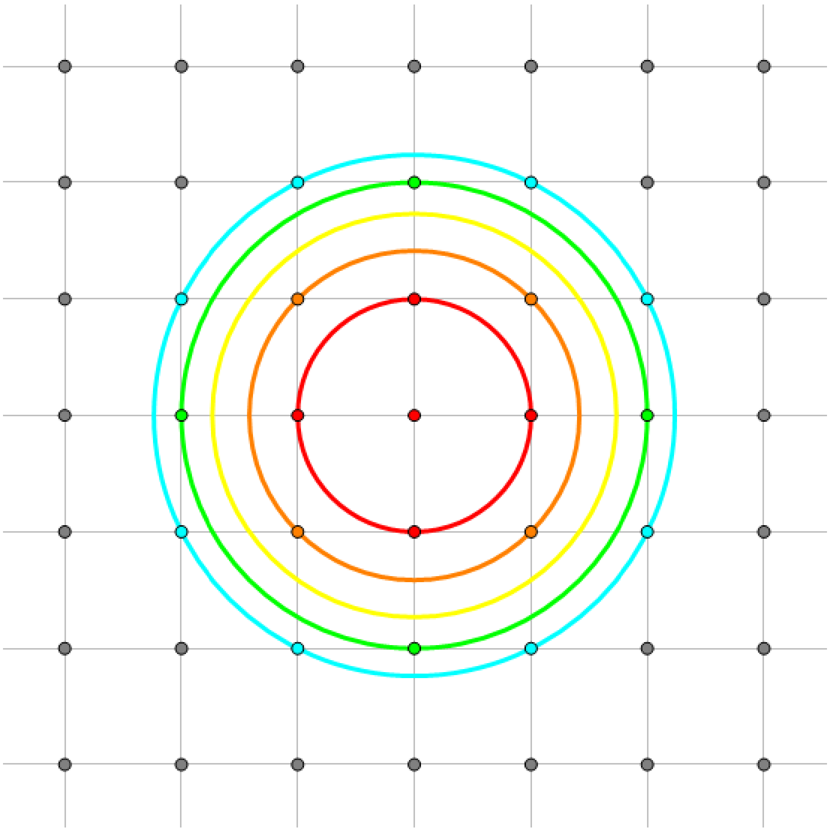
Examples

input
5
output
387

input
233
output
788243189

Note

A piece of paper with 5 circles is shown in the following.



There are 5 types of lattice points where the degree of beauty of each red point is $1+2+3+4+5=15$, the degree of beauty of each orange point is $2+3+4+5=14$, the degree of beauty of each green point is $4+5=9$, the degree of beauty of each blue point is 5 and the degree of beauty of each gray point is 0. Therefore, $f(5)=5\cdot 15+4\cdot 14+4\cdot 9+8\cdot 5=207$.

Similarly, $f(1)=5, f(2)=23, f(3)=50, f(4)=102$ and consequently $\left(\sum_{k=1}^{5} f(k)\right) \bmod (10^9+7) = 387$.

E. A Preponderant Reunion

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

East or west, home is best. That's why family reunion, the indispensable necessity of Lunar New Year celebration, is put in such a position.

After the reunion dinner, Little Tommy plays a game with the family. Here is a concise introduction to this game:

1. There is a sequence of n non-negative integers $p_1, p_2, \ldots, p_n$ in the beginning. It is ruled that each integer in this sequence should be non-negative **at any time**.
2. You can select two **consecutive positive** integers in this sequence, p_i and p_{i+1} $(1 \le i < n)$, and then decrease them by their minimum (i. e. $min(p_i, p_{i+1})$), the cost of this operation is equal to $min(p_i, p_{i+1})$. We call such operation as a *descension*.
3. The game immediately ends when there are no two consecutive positive integers. Your task is to end the game so that the total cost of your operations is as small as possible.

Obviously, every game ends after at most $n-1$ descensions. Please share your solution of this game with the lowest cost.

Input

The first line contains one integer n $(1 \le n \le 3\cdot 10^5)$.

The second line contains n space-separated integers $p_1, p_2, \ldots, p_n$ $(0 \le p_i \le 10^9, i=1,2,\ldots,n)$.

Output

In the first line print one integer as the number of descensions m $(0 \le m \le n-1)$.

In the next m lines print the descensions chronologically. More precisely, in each line of the next m lines print one integer i $(1 \le i < n)$ representing a descension would operate on p_i and p_{i+1} such that all the descensions could be utilized from top to bottom.

If there are many possible solutions to reach the minimal cost, print any of them.

Examples

input

```
4
2 1 3 1
```

output

```
2
1
3
```

input

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

output

```
3
2
1
4
```

Note

In the first sample, one possible best solution is $[2, 1, 3, 1] \to [1, 0, 3, 1] \to [1, 0, 2, 0]$, of which the cost is $1 + 1 = 2$.

In the second sample, one possible best solution is $[2, 2, 1, 3, 1] \to [2, 1, 0, 3, 1] \to [1, 0, 0, 3, 1] \to [1, 0, 0, 2, 0]$, of which the cost is $1 + 1 + 1 = 3$.