Codeforces Testing Round #2

A. Measuring Lengths in Baden

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

Lengths are measures in Baden in inches and feet. To a length from centimeters it is enough to know that an inch equals three centimeters in Baden and one foot contains 12 inches.

You are given a length equal to n centimeters. Your task is to convert it to feet and inches so that the number of feet was maximum. The result should be an integer rounded to the closest value containing an integral number of inches.

Note that when you round up, 1 cm rounds up to 0 inches and 2 cm round up to 1 inch.

Input
The only line contains an integer n ($1 \le n \le 10000$).

Output
Print two non-negative space-separated integers a and b, where a is the numbers of feet and b is the number of inches.

Sample test(s)

input
42
output
1 2

input
5
output
0 2

B. Simple XML

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

Let's define a string <x> as an opening tag, where x is any small letter of the Latin alphabet. Each opening tag matches a closing tag of the type </x>, where x is the same letter.

Tegs can be nested into each other: in this case one opening and closing tag pair is located inside another pair.

Let's define the notion of a *XML-text*:

- an empty string is a *XML-text*
- if s is a *XML-text*, then s'=<a>+s+ also is a *XML-text*, where a is any small Latin letter
- if s_1, s_2 are *XML-texts*, then s_1+s_2 also is a *XML-text*

You are given a XML-text (it is guaranteed that the text is valid), your task is to print in the following form:

- each tag (opening and closing) is located on a single line
- print before the tag $2 * h$ spaces, where h is the level of the tag's nestedness.

Input

The input data consists on the only non-empty string — the XML-text, its length does not exceed 1000 characters. It is guaranteed that the text is valid. The text contains no spaces.

Output

Print the given XML-text according to the above-given rules.

Sample test(s)

input

```
<a><b><c></c></b></a>
```

output

```
<a>
  <b>
    <c>
    </c>
  </b>
</a>
```

input

```
<a><b></b><d><c></c></d></a>
```

output

```
<a>
  <b>
  </b>
  <d>
    <c>
    </c>
  </d>
</a>
```

C. Hobbits' Party

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

Everyone knows that hobbits love to organize all sorts of parties and celebrations. There are n hobbits living in the Shire. They decided to organize the Greatest Party (GP) that would last for several days. Next day the hobbits wrote a guest list, some non-empty set containing all the inhabitants of the Shire. To ensure that everybody enjoy themselves and nobody gets bored, for any two days (say, days A and B) of the GP there existed at least one hobbit, invited to come on day A and on day B. However, to ensure that nobody has a row, for any three different days A, B, C there shouldn't be a hobbit invited on days A, B and C. The Shire inhabitants are keen on keeping the GP going for as long as possible. Your task is given number n, to indicate the GP's maximum duration and the guest lists for each day.

Input

The first line contains an integer n $(3 \le n \le 10000)$, representing the number of hobbits.

Output

In the first output line print a number k — the maximum duration of GP in days. Then on k lines print the guest lists, (the guests should be separated by spaces). Print each guest list on the single line. Each list can contain an arbitrary positive number of hobbits. The hobbits are numbered with integers from 1 to n.

Sample test(s)

input
4
output
3 1 2 1 3 2 3

input
5
output
3 1 2 1 3 2 3

D. Two progressions

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

An arithmetic progression is such a **non-empty** sequence of numbers where the difference between any two successive numbers is constant. This constant number is called common difference. For example, the sequence 3, 7, 11, 15 is an arithmetic progression. The definition implies that any sequences whose length equals 1 or 2 are arithmetic and all sequences whose length equals 0 are non-arithmetic.

You are given a sequence of **different** integers $a_1, a_2, ..., a_n$. You should either split it into two arithmetic progressions or find out that the operation is impossible to perform. Splitting assigns each member of the given sequence to one of two progressions, but the relative order of numbers does not change. Splitting is an inverse operation to *merging*.

Input

The first line contains a positive integer n ($2 \le n \le 30000$), n is the length of the given sequence. The second line contains elements of the given sequence $a_1, a_2, ..., a_n$ ($-10^8 \le a_i \le 10^8$). The elements of the progression are different integers.

Output

Print the required arithmetic progressions, one per line. The progressions can be positioned in any order. Each progression should contain at least one number. If there's no solution, then print "No solution" (without the quotes)in the only line of the input file. If there are several solutions, print any of them.

Sample test(s)

input
6 4 1 2 7 3 10
output
1 2 3 4 7 10

input
5 1 2 3 -2 -7
output
1 2 3 -2 -7

Note

In the second sample another solution is also possible (number three can be assigned to the second progression): 1, 2 and 3, -2, -7.

E. MST Company

time limit per test: 8 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The MST (Meaningless State Team) company won another tender for an important state reform in Berland.

There are n cities in Berland, some pairs of the cities are connected by roads. Each road has its price. One can move along any road in any direction. The MST team should carry out the repair works on some set of roads such that one can get from any city to any other one moving only along the repaired roads. Moreover, this set should contain exactly k capital roads (that is, the roads that start or finish in the capital). The number of the capital is 1.

As the budget has already been approved, the MST Company will profit by finding the set with minimum lengths of roads.

Input

The first input line contains three integers n, m, k ($1 \le n \le 5000; 0 \le m \le 10^5; 0 \le k < 5000$), where n is the number of cities in the country, m is the number of roads in the country, k is the number of capital roads in the required set. Then m lines enumerate the roads in question. Each road is specified by three numbers a_i, b_i, w_i ($1 \le a_i, b_i \le n; 1 \le w \le 10^5$), where a_i, b_i are the numbers of cities linked by a road and w_i is its length.

Between each pair of cities no more than one road exists. There are no roads that start and finish in one city. The capital's number is 1.

Output

In the first line print the number of roads in the required set. The second line should contain the numbers of roads included in the sought set. If the sought set does not exist, print -1.

Sample test(s)

input
4 5 2 1 2 1 2 3 1 3 4 1 1 3 3 1 4 2
output
3 1 5 2