Codeforces Beta Round #64

A. Cookies

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

Fangy collects cookies. Once he decided to take a box and put cookies into it in some way. If we take a square $k \times k$ in size, divided into blocks 1×1 in size and paint there the main diagonal together with cells, which lie above it, then the painted area will be equal to the area occupied by one cookie k in size. Fangy also has a box with a square base $2^n \times 2^n$, divided into blocks 1×1 in size. In a box the cookies should not overlap, and they should not be turned over or rotated. See cookies of sizes 2 and 4 respectively on the figure:

To stack the cookies the little walrus uses the following algorithm. He takes out of the repository the largest cookie which can fit in some place in the box and puts it there. Everything could be perfect but alas, in the repository the little walrus has infinitely many cookies of size 2 and larger, and there are no cookies of size 1, therefore, empty cells will remain in the box. Fangy wants to know how many empty cells will be left in the end.

Input
The first line contains a single integer n ($0 \le n \le 1000$).

Output
Print the single number, equal to the number of empty cells in the box. The answer should be printed modulo $10^6 + 3$.

Examples

input
3
output
9

Note
If the box possesses the base of $2^3 \times 2^3$ (as in the example), then the cookies will be put there in the following manner:

B. Text Messaging

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

Fangy the little walrus, as all the modern walruses, loves to communicate via text messaging. One day he faced the following problem: When he sends large texts, they are split into parts each containing n characters (which is the size of one text message). Thus, whole sentences and words get split!

Fangy did not like it, so he faced the task of breaking the text into minimal messages on his own so that no sentence were broken into pieces when it is sent and the number of text messages to be sent would be minimal. If two consecutive sentences are in different messages, the space between them can be ignored (Fangy does not write this space).

The little walrus's text looks in the following manner:

```
TEXT ::= SENTENCE | SENTENCE SPACE TEXT
SENTENCE ::= WORD SPACE SENTENCE | WORD END
END ::= {'.', '?', '!'}
WORD ::= LETTER | LETTER WORD
LETTER ::= {'a'..'z', 'A'..'Z'}
SPACE ::= ' '
```

SPACE stands for the symbol of a space.

So, how many messages did Fangy send?

Input

The first line contains an integer n, which is the size of one message ($2 \le n \le 255$). The second line contains the text. The length of the text does not exceed 10^4 characters. It is guaranteed that the text satisfies the above described format. Specifically, this implies that the text is not empty.

Output

On the first and only line print the number of text messages Fangy will need. If it is impossible to split the text, print "`Impossible`" without the quotes.

Examples

input
25 Hello. I am a little walrus.
output
2

input
2 How are you?
output
Impossible

input
19 Hello! Do you like fish? Why?
output
3

Note

Let's take a look at the third sample. The text will be split into three messages: "`Hello!`", "`Do you like fish?`" and "`Why?`".

C. Lucky Tickets

time limit per test: 1.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In Walrusland public transport tickets are characterized by two integers: by the number of the series and by the number of the ticket in the series. Let the series number be represented by a and the ticket number — by b, then a ticket is described by the ordered pair of numbers (a, b).

The walruses believe that a ticket is lucky if $a * b = rev(a) * rev(b)$. The function $rev(x)$ reverses a number written in the decimal system, at that the leading zeroes disappear. For example, $rev(12343) = 34321$, $rev(1200) = 21$.

The Public Transport Management Committee wants to release x series, each containing y tickets, so that **at least** w lucky tickets were released and the total number of released tickets ($x * y$) were minimum. The series are numbered from 1 to x inclusive. The tickets in each series are numbered from 1 to y inclusive. The Transport Committee cannot release more than max_x series and more than max_y tickets in one series.

Input

The first line contains three integers max_x, max_y, w ($1 \le max_x, max_y \le 10^5$, $1 \le w \le 10^7$).

Output

Print on a single line two space-separated numbers, the x and the y. If there are several possible variants, print any of them. If such x and y do not exist, print a single number -1.

Examples

input
2 2 1
output
1 1

input
132 10 35
output
7 5

input
5 18 1000
output
-1

input
48 132 235
output
22 111

D. Professor's task

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

Once a walrus professor Plato asked his programming students to perform the following practical task.

The students had to implement such a data structure that would support a convex hull on some set of points S. The input to the program had q queries of two types:

1. Add a point with coordinates (x, y) into the set S. Note that in this case the convex hull of S could have changed, and could have remained the same.

2. Say whether a point with coordinates (x, y) belongs to an area limited by the convex hull, including the border.

All the students coped with the task. What about you?

Input

The first line contains an integer q ($4 \le q \le 10^5$).

Then follow q lines in the following way: "$t\ x\ y$", where t is the query type (1 or 2), and (x, y) are the coordinates of the point ($-10^6 \le x, y \le 10^6$, x and y are integers).

There is at least one query of type 2.

It is guaranteed that the three queries of the first type follow first and the points given in the queries form a non-degenerative triangle. Also all the points added in S are distinct.

Output

For each query of the second type print one string containing "YES", if the point lies inside the convex hull or on its border. Otherwise, print "NO".

Examples

input
8 1 0 0 1 2 0 1 2 2 2 1 0 1 0 2 2 1 1 2 2 1 2 20 -1
output
YES YES YES NO

E. Information Reform

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

Thought it is already the XXI century, the Mass Media isn't very popular in Walrusland. The cities get news from messengers who can only travel along roads. The network of roads in Walrusland is built so that it is possible to get to any city from any other one in exactly one way, and the roads' lengths are equal.

The North Pole governor decided to carry out an information reform. Several cities were decided to be chosen and made regional centers. Maintaining a region center takes k fishlars (which is a local currency) per year. It is assumed that a regional center always has information on the latest news.

For every city which is not a regional center, it was decided to appoint a regional center which will be responsible for keeping this city informed. In that case the maintenance costs will be equal to d_{len} fishlars per year, where len is the distance from a city to the corresponding regional center, measured in the number of roads along which one needs to go.

Your task is to minimize the costs to carry out the reform.

Input

The first line contains two given numbers n and k ($1 \le n \le 180, 1 \le k \le 10^5$).

The second line contains $n - 1$ integers d_i, numbered starting with 1 ($d_i \le d_{i+1}, 0 \le d_i \le 10^5$).

Next $n - 1$ lines contain the pairs of cities connected by a road.

Output

On the first line print the minimum number of fishlars needed for a year's maintenance. On the second line print n numbers, where the i-th number will represent the number of the regional center, appointed to the i-th city. If the i-th city is a regional center itself, then you should print number i.

If there are several solutions to that problem, print any of them.

Examples

input
8 10 2 5 9 11 15 19 20 1 4 1 3 1 7 4 6 2 8 2 3 3 5

output
38 3 3 3 4 3 4 3 3