Codeforces Round #181 (Div. 2)

A. Array

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

Vitaly has an array of n distinct integers. Vitaly wants to divide this array into three **non-empty** sets so as the following conditions hold:

1. The product of all numbers in the first set is less than zero (< 0).
2. The product of all numbers in the second set is greater than zero (> 0).
3. The product of all numbers in the third set is equal to zero.
4. Each number from the initial array must occur in exactly one set.

Help Vitaly. Divide the given array.

Input

The first line of the input contains integer n $(3 \le n \le 100)$. The second line contains n space-separated distinct integers $a_1, a_2, ..., a_n$ $(|a_i| \le 10^3)$ — the array elements.

Output

In the first line print integer n_1 $(n_1 > 0)$ — the number of elements in the first set. Then print n_1 numbers — the elements that got to the first set.

In the next line print integer n_2 $(n_2 > 0)$ — the number of elements in the second set. Then print n_2 numbers — the elements that got to the second set.

In the next line print integer n_3 $(n_3 > 0)$ — the number of elements in the third set. Then print n_3 numbers — the elements that got to the third set.

The printed sets must meet the described conditions. It is guaranteed that the solution exists. If there are several solutions, you are allowed to print any of them.

Sample test(s)

input

```
3
-1 2 0
```

output

```
1 -1
1 2
1 0
```

input

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

output

```
1 -1
2 -3 -2
1 0
```

B. Coach

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

A programming coach has n students to teach. We know that n is divisible by 3. Let's assume that all students are numbered from 1 to n, inclusive.

Before the university programming championship the coach wants to split all students into groups of three. For some pairs of students we know that they want to be on the same team. Besides, if the i-th student wants to be on the same team with the j-th one, then the j-th student wants to be on the same team with the i-th one. The coach wants the teams to show good results, so he wants the following condition to hold: if the i-th student wants to be on the same team with the j-th, then the i-th and the j-th students must be on the same team. Also, it is obvious that each student must be on exactly one team.

Help the coach and divide the teams the way he wants.

Input

The first line of the input contains integers n and m $(3 \le n \le 48, 0 \le m \le \frac{n(n-1)}{2})$. Then follow m lines, each contains a pair of integers a_i, b_i $(1 \le a_i < b_i \le n)$ — the pair a_i, b_i means that students with numbers a_i and b_i want to be on the same team.

It is guaranteed that n is divisible by 3. It is guaranteed that each pair a_i, b_i occurs in the input at most once.

Output

If the required division into teams doesn't exist, print number `-1`. Otherwise, print $\frac{n}{3}$ lines. In each line print three integers x_i, y_i, z_i $(1 \le x_i, y_i, z_i \le n)$ — the i-th team.

If there are multiple answers, you are allowed to print any of them.

Sample test(s)

input

```
3 0
```

output

```
3 2 1
```

input

```
6 4
1 2
2 3
3 4
5 6
```

output

```
-1
```

input

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

output

```
3 2 1
```

C. Beautiful Numbers

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

Vitaly is a very weird man. He's got two favorite digits a and b. Vitaly calls a positive integer *good*, if the decimal representation of this integer only contains digits a and b. Vitaly calls a good number *excellent*, if the sum of its digits is a good number.

For example, let's say that Vitaly's favourite digits are 1 and 3, then number 12 isn't good and numbers 13 or 311 are. Also, number 111 is excellent and number 11 isn't.

Now Vitaly is wondering, how many excellent numbers of length exactly n are there. As this number can be rather large, he asks you to count the remainder after dividing it by 1000000007 $(10^9 + 7)$.

A number's length is the number of digits in its decimal representation without leading zeroes.

Input

The first line contains three integers: a, b, n $(1 \le a < b \le 9, 1 \le n \le 10^6)$.

Output

Print a single integer — the answer to the problem modulo 1000000007 $(10^9 + 7)$.

Sample test(s)

input

```
1 3 3
```

output

```
1
```

input

```
2 3 10
```

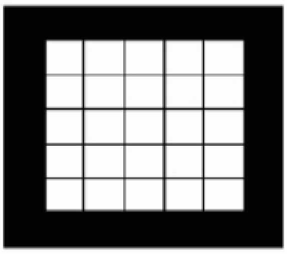
output

```
165
```

D. Painting Square

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

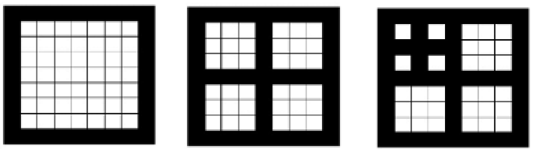
Vasily the bear has got a large square white table of n rows and n columns. The table has got a black border around this table.



The example of the initial table at n = 5.

Vasily the bear wants to paint his square table in exactly k moves. Each move is sequence of actions:

1. The bear chooses some square inside his table. At that the square must have a black border painted around it. Also, the square shouldn't contain a black cell. The number of cells in the square shouldn't be less than 2.
2. The bear chooses some row and some column inside the chosen square. Then he paints each cell of this row and this column inside the chosen square. After that the rectangles, formed by the square's border and the newly painted cells, must be squares of a non-zero area.



An example of correct painting at n = 7 и k = 2.

The bear already knows numbers n and k. Help him — find the number of ways to paint the square in exactly k moves. Two ways to paint are called distinct if the resulting tables will differ in at least one cell. As the answer can be rather large, print the remainder after dividing it by 7340033.

Input

The first line contains integer q $(1 \le q \le 10^5)$ — the number of test data.

Each of the following q lines contains two integers n and k $(1 \le n \le 10^9, 0 \le k \le 1000)$ — the size of the initial table and the number of moves for the corresponding test.

Output

For each test from the input print the answer to the problem modulo 7340033. Print the answers to the tests in the order in which the tests are given in the input.

Sample test(s)

input

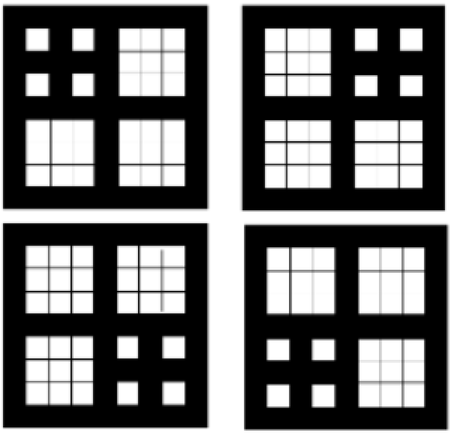
```
8
1 0
1 1
3 0
3 1
2 0
2 1
3 2
7 2
```

output

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

Note

All possible painting ways for the test $n = 7$ and $k = 2$ are:



E. Empire Strikes Back

time limit per test: 5 seconds

memory limit per test: 512 megabytes

input: standard input

output: standard output

In a far away galaxy there is war again. The treacherous Republic made k precision strikes of power a_i on the Empire possessions. To cope with the republican threat, the Supreme Council decided to deal a decisive blow to the enemy forces.

To successfully complete the conflict, the confrontation balance after the blow should be a positive integer. The balance of confrontation is a number that looks like $\frac{p}{q}$, where $p = n!$ (n is the power of the Imperial strike), $q = \prod_{i=1}^{k} a_i!$. After many years of war the Empire's resources are low. So to reduce the costs, n should be a minimum positive integer that is approved by the commanders.

Help the Empire, find the minimum positive integer n, where the described fraction is a positive integer.

Input

The first line contains integer k $(1 \le k \le 10^6)$. The second line contains k integers $a_1, a_2, ..., a_k$ $(1 \le a_i \le 10^7)$.

Output

Print the minimum positive integer n, needed for the Empire to win.

Please, do not use the `%lld` to read or write 64-but integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specificator.

Sample test(s)

input

```
2
1000 1000
```

output

```
2000
```

input

```
1
2
```

output

```
2
```