A. Olesya and Rodion

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

Olesya loves numbers consisting of n digits, and Rodion only likes numbers that are divisible by t. Find some number that satisfies both of them.

Your task is: given the n and t print an integer strictly larger than zero consisting of n digits that is divisible by t. If such number doesn't exist, print -1.

Input

The single line contains two numbers, n and t ($1 \le n \le 100$, $2 \le t \le 10$) — the length of the number and the number it should be divisible by.

Output

Print one such positive number without leading zeroes, — the answer to the problem, or -1, if such number doesn't exist. If there are multiple possible answers, you are allowed to print any of them.

Sample test(s)

input

```
3 2
```

output

```
712
```

B. Kolya and Tanya

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

Kolya loves putting gnomes at the circle table and giving them coins, and Tanya loves studying triplets of gnomes, sitting in the vertexes of an equilateral triangle.

More formally, there are $3n$ gnomes sitting in a circle. Each gnome can have from 1 to 3 coins. Let's number the places in the order they occur in the circle by numbers from 0 to $3n - 1$, let the gnome sitting on the i-th place have a_i coins. If there is an integer i ($0 \le i < n$) such that $a_i + a_{i+n} + a_{i+2n} \ne 6$, then Tanya is satisfied.

Count the number of ways to choose a_i so that Tanya is satisfied. As there can be many ways of distributing coins, print the remainder of this number modulo $10^9 + 7$. Two ways, a and b, are considered distinct if there is index i ($0 \le i < 3n$), such that $a_i \ne b_i$ (that is, some gnome got different number of coins in these two ways).

Input

A single line contains number n ($1 \le n \le 10^5$) — the number of the gnomes divided by three.

Output

Print a single number — the remainder of the number of variants of distributing coins that satisfy Tanya modulo $10^9 + 7$.

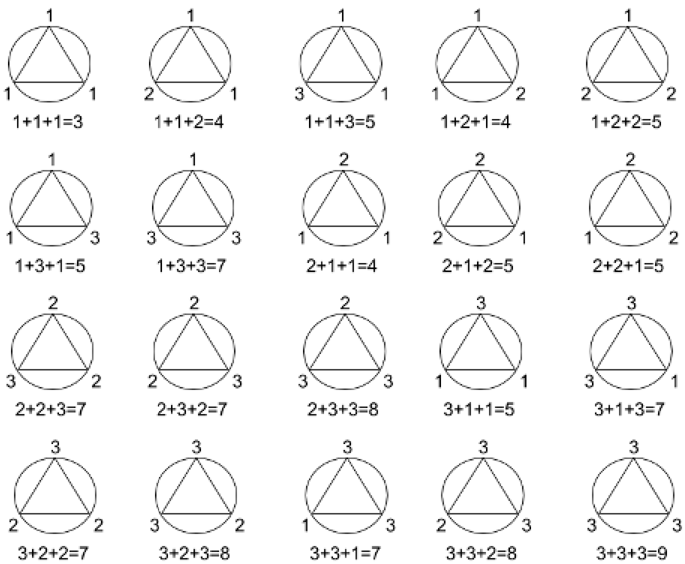
Sample test(s)

```
input
1
output
20
```

```
input
2
output
680
```

Note

20 ways for $n = 1$ (gnome with index 0 sits on the top of the triangle, gnome 1 on the right vertex, gnome 2 on the left vertex):



C. Marina and Vasya

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

Marina loves strings of the same length and Vasya loves when there is a third string, different from them in exactly t characters. Help Vasya find at least one such string.

More formally, you are given two strings s_1, s_2 of length n and number t. Let's denote as $f(a, b)$ the number of characters in which strings a and b are different. Then your task will be to find any string s_3 of length n, such that $f(s_1, s_3) = f(s_2, s_3) = t$. If there is no such string, print -1.

Input

The first line contains two integers n and t ($1 \le n \le 10^5$, $0 \le t \le n$).

The second line contains string s_1 of length n, consisting of lowercase English letters.

The third line contain string s_2 of length n, consisting of lowercase English letters.

Output

Print a string of length n, differing from string s_1 and from s_2 in exactly t characters. Your string should consist only from lowercase English letters. If such string doesn't exist, print -1.

Sample test(s)

input

```
3 2
abc
xyc
```

output

```
ayd
```

input

```
1 0
c
b
```

output

```
-1
```

D. Dima and Lisa

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

Dima loves representing an odd number as the sum of multiple primes, and Lisa loves it when there are at most three primes. Help them to represent the given number as the sum of at most than three primes.

More formally, you are given an **odd** numer n. Find a set of numbers p_i $(1 \le i \le k)$, such that

1. $1 \le k \le 3$
2. p_i is a prime
3. $\sum_{i=1}^{k} p_i = n$

The numbers p_i do not necessarily have to be distinct. It is guaranteed that at least one possible solution exists.

Input

The single line contains an odd number n $(3 \le n < 10^9)$.

Output

In the first line print k $(1 \le k \le 3)$, showing how many numbers are in the representation you found.

In the second line print numbers p_i in any order. If there are multiple possible solutions, you can print any of them.

Sample test(s)

input

```
27
```

output

```
3
5 11 11
```

Note

A prime is an integer strictly larger than one that is divisible only by one and by itself.

E. Anton and Ira

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

Anton loves transforming one permutation into another one by swapping elements for money, and Ira doesn't like paying for stupid games. Help them obtain the required permutation by paying as little money as possible.

More formally, we have two permutations, p and s of numbers from 1 to n. We can swap p_i and p_j, by paying $|i - j|$ coins for it. Find and print the smallest number of coins required to obtain permutation s from permutation p. Also print the sequence of swap operations at which we obtain a solution.

Input

The first line contains a single number n $(1 \leq n \leq 2000)$ — the length of the permutations.

The second line contains a sequence of n numbers from 1 to n — permutation p. Each number from 1 to n occurs exactly once in this line.

The third line contains a sequence of n numbers from 1 to n — permutation s. Each number from 1 to n occurs once in this line.

Output

In the first line print the minimum number of coins that you need to spend to transform permutation p into permutation s.

In the second line print number k $(0 \leq k \leq 2 \cdot 10^6)$ — the number of operations needed to get the solution.

In the next k lines print the operations. Each line must contain two numbers i and j $(1 \leq i, j \leq n, i \neq j)$, which means that you need to swap p_i and p_j.

It is guaranteed that the solution exists.

Sample test(s)

input

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

output

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

Note

In the first sample test we swap numbers on positions 3 and 4 and permutation p becomes 4 2 3 1. We pay $|3 - 4| = 1$ coins for that. On second turn we swap numbers on positions 1 and 3 and get permutation 3241 equal to s. We pay $|3 - 1| = 2$ coins for that. In total we pay three coins.