A. Levko and Table

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

Levko loves tables that consist of n rows and n columns very much. He especially loves beautiful tables. A table is *beautiful* to Levko if the sum of elements in each row and column of the table equals k.

Unfortunately, he doesn't know any such table. Your task is to help him to find at least one of them.

Input

The single line contains two integers, n and k ($1 \le n \le 100$, $1 \le k \le 1000$).

Output

Print any beautiful table. Levko doesn't like too big numbers, so all elements of the table mustn't exceed 1000 in their absolute value.

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

Examples

input

```
2 4
```

output

```
1 3
3 1
```

input

```
4 7
```

output

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

Note

In the first sample the sum in the first row is $1+3=4$, in the second row — $3+1=4$, in the first column — $1+3=4$ and in the second column — $3+1=4$. There are other beautiful tables for this sample.

In the second sample the sum of elements in each row and each column equals 7. Besides, there are other tables that meet the statement requirements.

B. Levko and Permutation

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

Levko loves permutations very much. A permutation of length n is a sequence of distinct positive integers, each is at most n.

Let's assume that value $gcd(a, b)$ shows the greatest common divisor of numbers a and b. Levko assumes that element p_i of permutation $p_1, p_2, \dots, p_n$ is good if $gcd(i, p_i) > 1$. Levko considers a permutation *beautiful*, if it has exactly k good elements. Unfortunately, he doesn't know any beautiful permutation. Your task is to help him to find at least one of them.

Input

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

Output

In a single line print either any beautiful permutation or -1, if such permutation doesn't exist.

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

Examples

input

```
4 2
```

output

```
2 4 3 1
```

input

```
1 1
```

output

```
-1
```

Note

In the first sample elements 4 and 3 are good because $gcd(2, 4) = 2 > 1$ and $gcd(3, 3) = 3 > 1$. Elements 2 and 1 are not good because $gcd(1, 2) = 1$ and $gcd(4, 1) = 1$. As there are exactly 2 good elements, the permutation is beautiful.

The second sample has no beautiful permutations.

C. Levko and Array Recovery

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

Levko loves array $a_1, a_2, \dots, a_n$, consisting of integers, very much. That is why Levko is playing with array a, performing all sorts of operations with it. Each operation Levko performs is of one of two types:

1. Increase all elements from l_i to r_i by d_i. In other words, perform assignments $a_j = a_j + d_i$ for all j that meet the inequation $l_i \le j \le r_i$.
2. Find the maximum of elements from l_i to r_i. That is, calculate the value $m_i = \max_{j=l_i}^{r_i} a_j$.

Sadly, Levko has recently lost his array. Fortunately, Levko has records of all operations he has performed on array a. Help Levko, given the operation records, find at least one suitable array. The results of all operations for the given array must coincide with the record results. Levko clearly remembers that all numbers in his array didn't exceed 10^9 in their absolute value, so he asks you to find such an array.

Input

The first line contains two integers n and m ($1 \le n, m \le 5000$) — the size of the array and the number of operations in Levko's records, correspondingly.

Next m lines describe the operations, the i-th line describes the i-th operation. The first integer in the i-th line is integer t_i ($1 \le t_i \le 2$) that describes the operation type. If $t_i = 1$, then it is followed by three integers l_i, r_i and d_i ($1 \le l_i \le r_i \le n$, $-10^4 \le d_i \le 10^4$) — the description of the operation of the first type. If $t_i = 2$, then it is followed by three integers l_i, r_i and m_i ($1 \le l_i \le r_i \le n$, $-5 \cdot 10^7 \le m_i \le 5 \cdot 10^7$) — the description of the operation of the second type.

The operations are given in the order Levko performed them on his array.

Output

In the first line print "YES" (without the quotes), if the solution exists and "NO" (without the quotes) otherwise.

If the solution exists, then on the second line print n integers $a_1, a_2, \dots, a_n$ ($|a_i| \le 10^9$) — the recovered array.

Examples

input

```
4 5
1 2 3 1
2 1 2 8
2 3 4 7
1 1 3 3
2 3 4 8
```

output

```
YES
4 7 4 7
```

input

```
4 5
1 2 3 1
2 1 2 8
2 3 4 7
1 1 3 3
2 3 4 13
```

output

```
NO
```

D. Levko and Array

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

Levko has an array that consists of integers: $a_1, a_2, \ldots, a_n$. But he doesn't like this array at all.

Levko thinks that the beauty of the array a directly depends on value $c(a)$, which can be calculated by the formula:

$$c(a) = \max_{1 \le i \le n-1} |a_{i+1} - a_i|, n > 1; c(a) = 0, 0 \le n \le 1.$$

The less value $c(a)$ is, the more beautiful the array is.
It's time to change the world and Levko is going to change his array for the better. To be exact, Levko wants to change the values of at most k array elements (it is allowed to replace the values by any integers). Of course, the changes should make the array as beautiful as possible.

Help Levko and calculate what minimum number $c(a)$ he can reach.

Input

The first line contains two integers n and k ($1 \le k \le n \le 2000$). The second line contains space-separated integers $a_1, a_2, \ldots, a_n$ ($-10^9 \le a_i \le 10^9$).

Output

A single number — the minimum value of $c(a)$ Levko can get.

Examples

input

```
5 2
4 7 4 7 4
```

output

```
0
```

input

```
3 1
-100 0 100
```

output

```
100
```

input

```
6 3
1 2 3 7 8 9
```

output

```
1
```

Note

In the first sample Levko can change the second and fourth elements and get array: $4, 4, 4, 4, 4$.

In the third sample he can get array: $1, 2, 3, 4, 5, 6$.

E. Levko and Strings

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

Levko loves strings of length n, consisting of lowercase English letters, very much. He has one such string s. For each string t of length n, Levko defines its beauty relative to s as the number of pairs of indexes i, j $(1 \le i \le j \le n)$, such that substring **$t[i..j]$ is lexicographically larger than substring $s[i..j]$**.

The boy wondered how many strings t are there, such that their beauty relative to s equals exactly k. Help him, find the remainder after division this number by 1000000007 $(10^9 + 7)$.

A substring $s[i..j]$ of string $s = s_1s_2... s_n$ is string $s_is_{i+1}... s_j$.

String $x = x_1x_2... x_p$ is lexicographically larger than string $y = y_1y_2... y_p$, if there is such number r ($r < p$), that $x_1 = y_1, x_2 = y_2, ... , x_r = y_r$ and $x_{r+1} > y_{r+1}$. The string characters are compared by their ASCII codes.

Input

The first line contains two integers n and k ($1 \le n \le 2000$, $0 \le k \le 2000$).

The second line contains a non-empty string s of length n. String s consists only of lowercase English letters.

Output

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

Examples

input

```
2 2
yz
```

output

```
26
```

input

```
2 3
yx
```

output

```
2
```

input

```
4 7
abcd
```

output

```
21962
```