A. Knight Tournament

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

Hooray! Berl II, the king of Berland is making a knight tournament. The king has already sent the message to all knights in the kingdom and they in turn agreed to participate in this grand event.

As for you, you're just a simple peasant. There's no surprise that you slept in this morning and were late for the tournament (it was a weekend, after all). Now you are really curious about the results of the tournament. This time the tournament in Berland went as follows:

- There are n knights participating in the tournament. Each knight was assigned his unique number — an integer from 1 to n.
- The tournament consisted of m fights, in the i-th fight the knights that were still in the game with numbers at least l_i and at most r_i have fought for the right to continue taking part in the tournament.
- After the i-th fight among all participants of the fight only one knight won — the knight number x_i, he continued participating in the tournament. Other knights left the tournament.
- The winner of the last (the m-th) fight (the knight number x_m) became the winner of the tournament.

You fished out all the information about the fights from your friends. Now for each knight you want to know the name of the knight he was conquered by. We think that the knight number b was conquered by the knight number a, if there was a fight with both of these knights present and the winner was the knight number a.

Write the code that calculates for each knight, the name of the knight that beat him.

Input

The first line contains two integers n, m $(2 \le n \le 3 \cdot 10^5;\ 1 \le m \le 3 \cdot 10^5)$ — the number of knights and the number of fights. Each of the following m lines contains three integers l_i, r_i, x_i $(1 \le l_i < r_i \le n;\ l_i \le x_i \le r_i)$ — the description of the i-th fight.

It is guaranteed that the input is correct and matches the problem statement. It is guaranteed that at least two knights took part in each battle.

Output

Print n integers. If the i-th knight lost, then the i-th number should equal the number of the knight that beat the knight number i. If the i-th knight is the winner, then the i-th number must equal 0.

Examples

input

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

output

```
3 1 4 0
```

input

```
8 4
3 5 4
3 7 6
2 8 8
1 8 1
```

output

```
0 8 4 6 4 8 6 1
```

Note

Consider the first test case. Knights 1 and 2 fought the first fight and knight 1 won. Knights 1 and 3 fought the second fight and knight 3 won. The last fight was between knights 3 and 4, knight 4 won.

B. Xenia and Hamming

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

Xenia is an amateur programmer. Today on the IT lesson she learned about the Hamming distance.

The Hamming distance between two strings $s = s_1s_2... s_n$ and $t = t_1t_2... t_n$ of equal length n is value $\sum_{i=1}^{n}[s_i \neq t_i]$. Record $[s_i \neq t_i]$ is the Iverson notation and represents the following: if $s_i \neq t_i$, it is one, otherwise — zero.

Now Xenia wants to calculate the Hamming distance between two long strings a and b. The first string a is the concatenation of n copies of string x, that is, $a = x + x + \dots + x = \sum_{i=1}^{n} x$. The second string b is the concatenation of m copies of string y.

Help Xenia, calculate the required Hamming distance, given n, x, m, y.

Input

The first line contains two integers n and m $(1 \le n, m \le 10^{12})$. The second line contains a non-empty string x. The third line contains a non-empty string y. Both strings consist of at most 10^6 lowercase English letters.

It is guaranteed that strings a and b that you obtain from the input have the same length.

Output

Print a single integer — the required Hamming distance.

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

Examples

input

```
100 10
a
aaaaaaaaaa
```

output

```
0
```

input

```
1 1
abacaba
abzczzz
```

output

```
4
```

input

```
2 3
rzr
az
```

output

```
5
```

Note

In the first test case string a is the same as string b and equals 100 letters a. As both strings are equal, the Hamming distance between them is zero.

In the second test case strings a and b differ in their 3-rd, 5-th, 6-th and 7-th characters. Thus, the Hamming distance equals 4.

In the third test case string a is `rzrrzr` and string b is `azazaz`. The strings differ in all characters apart for the second one, the Hamming distance between them equals 5.

C. Compartments

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

A team of students from the city S is sent to the All-Berland Olympiad in Informatics. Traditionally, they go on the train. All students have bought tickets in one carriage, consisting of n compartments (each compartment has exactly four people). We know that if one compartment contain one or two students, then they get bored, and if one compartment contain three or four students, then the compartment has fun throughout the entire trip.

The students want to swap with other people, so that no compartment with students had bored students. To swap places with another person, you need to convince him that it is really necessary. The students can not independently find the necessary arguments, so they asked a sympathetic conductor for help. The conductor can use her life experience to persuade any passenger to switch places with some student.

However, the conductor does not want to waste time persuading the wrong people, so she wants to know what is the minimum number of people necessary to persuade her to change places with the students. Your task is to find the number.

After all the swaps each compartment should either have no student left, or have a company of three or four students.

Input

The first line contains integer n ($1 \le n \le 10^6$) — the number of compartments in the carriage. The second line contains n integers $a_1, a_2, \ldots, a_n$ showing how many students ride in each compartment ($0 \le a_i \le 4$). It is guaranteed that at least one student is riding in the train.

Output

If no sequence of swapping seats with other people leads to the desired result, print number "-1" (without the quotes). In another case, print the smallest number of people you need to persuade to swap places.

Examples

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

input
```
3
4 1 1
```
output
```
2
```

input
```
4
0 3 0 4
```
output
```
0
```

D. Bags and Coins

time limit per test: 2.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

When you were a child you must have been told a puzzle of bags and coins. Anyway, here's one of its versions:

A horse has three bags. The first bag has one coin, the second bag has one coin and the third bag has three coins. In total, the horse has three coins in the bags. How is that possible?

The answer is quite simple. The third bag contains a coin and two other bags.

This problem is a generalization of the childhood puzzle. You have n bags. You know that the first bag contains a_1 coins, the second bag contains a_2 coins, ..., the n-th bag contains a_n coins. In total, there are s coins. Find the way to arrange the bags and coins so that they match the described scenario or else state that it is impossible to do.

Input

The first line contains two integers n and s $(1 \le n, s \le 70000)$ — the number of bags and the total number of coins. The next line contains n integers $a_1, a_2, \dots, a_n$ $(1 \le a_i \le 70000)$, where a_i shows the number of coins in the i-th bag.

Output

If the answer doesn't exist, print -1.

Otherwise, print n lines, on the i-th line print the contents of the i-th bag. The first number in the line, c_i $(0 \le c_i \le a_i)$, must represent the number of coins lying directly in the i-th bag (the coins in the bags that are in the i-th bag are not taken into consideration). The second number in the line, k_i $(0 \le k_i < n)$ must represent the number of bags that lie directly in the i-th bag (the bags that are inside the bags lying in the i-th bag are not taken into consideration). Next, the line must contain k_i integers — the numbers of the bags that are lying directly in the i-th bag.

The total number of coins in the solution must equal s. If we count the total number of coins the i-th bag in the solution has, we should get a_i.

No bag can directly lie in more than one bag. The bags can be nested in more than one level (see the second test case). If there are multiple correct answers, you can print any of them.

Examples

input
```
3 3
1 3 1
```
output
```
1 0
1 2 3 1
1 0
```

input
```
3 3
1 3 1
```
output
```
1 0
2 1 3
0 1 1
```

input
```
1 2
1
```
output
```
-1
```

input
```
8 10
2 7 3 4 1 3 1 2
```
output
```
2 0
1 2 1 4
0 2 7 8
0 2 5 6
1 0
```

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

Note

The pictures below show two possible ways to solve one test case from the statement. The left picture corresponds to the first test case, the right picture corresponds to the second one.

E. Xenia and String Problem

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

Xenia the coder went to The Olympiad of Informatics and got a string problem. Unfortunately, Xenia isn't fabulous in string algorithms. Help her solve the problem.

String s is a sequence of characters $s_1s_2\dots s_{|s|}$, where record $|s|$ shows the length of the string.

Substring $s[i\dots j]$ of string s is string $s_is_{i+1}\dots s_j$.

String s is a *Gray* string, if it meets the conditions:

- the length of string $|s|$ is odd;
- character $s_{\frac{|s|+1}{2}}$ occurs exactly once in the string;
- either $|s| = 1$, or substrings $s[1\dots\frac{|s|+1}{2}-1]$ and $s[\frac{|s|+1}{2}+1\dots|s|]$ are the same and are Gray strings.

For example, strings "abacaba", "xzx", "g" are Gray strings and strings "aaa", "xz", "abaxcbc" are not.

The *beauty* of string p is the sum of the squares of the lengths of all substrings of string p that are Gray strings. In other words, consider all pairs of values i, j $(1 \le i \le j \le |p|)$. If substring $p[i\dots j]$ is a Gray string, you should add $(j - i + 1)^2$ to the beauty.

Xenia has got string t consisting of lowercase English letters. She is allowed to replace at most one letter of the string by any other English letter. The task is to get a string of maximum beauty.

Input

The first line contains a non-empty string t $(1 \le |t| \le 10^5)$. String t only consists of lowercase English letters.

Output

Print the sought maximum beauty value Xenia can get.

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

Examples

input
```
zzz
```
output
```
12
```

input
```
aba
```
output
```
12
```

input
```
abacaba
```
output
```
83
```

input
```
aaaaaa
```
output
```
15
```

Note

In the first test sample the given string can be transformed into string p = "zbz". Such string contains Gray strings as substrings $p[1\dots 1]$, $p[2\dots 2]$, $p[3\dots 3]$ и $p[1\dots 3]$. In total, the beauty of string p gets equal to $1^2 + 1^2 + 1^2 + 3^2 = 12$. You can't obtain a more beautiful string.

In the second test case it is not necessary to perform any operation. The initial string has the maximum possible beauty.