A. Fraction

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

Petya is a big fan of mathematics, especially its part related to fractions. Recently he learned that a fraction $\frac{a}{b}$ is called *proper* iff its numerator is smaller than its denominator ($a < b$) and that the fraction is called *irreducible* if its numerator and its denominator are coprime (they do not have positive common divisors except 1).

During his free time, Petya thinks about proper irreducible fractions and converts them to decimals using the calculator. One day he mistakenly pressed addition button ($+$) instead of division button ($\div$) and got sum of numerator and denominator that was equal to n instead of the expected decimal notation.

Petya wanted to restore the original fraction, but soon he realized that it might not be done uniquely. That's why he decided to determine maximum possible proper irreducible fraction $\frac{a}{b}$ such that sum of its numerator and denominator equals n. Help Petya deal with this problem.

Input

In the only line of input there is an integer n ($3 \le n \le 1000$), the sum of numerator and denominator of the fraction.

Output

Output two space-separated positive integers a and b, numerator and denominator of the maximum possible proper irreducible fraction satisfying the given sum.

Examples

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

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

input
```
12
```
output
```
5 7
```

B. Maxim Buys an Apartment

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

Maxim wants to buy an apartment in a new house at Line Avenue of Metropolis. The house has n apartments that are numbered from 1 to n and are arranged in a row. Two apartments are adjacent if their indices differ by 1. Some of the apartments can already be inhabited, others are available for sale.

Maxim often visits his neighbors, so apartment is *good* for him if it is available for sale and there is at least one already inhabited apartment adjacent to it. Maxim knows that there are exactly k already inhabited apartments, but he doesn't know their indices yet.

Find out what could be the minimum possible and the maximum possible number of apartments that are good for Maxim.

Input

The only line of the input contains two integers: n and k ($1 \le n \le 10^9$, $0 \le k \le n$).

Output

Print the minimum possible and the maximum possible number of apartments good for Maxim.

Example

input
```
6 3
```
output
```
1 3
```

Note

In the sample test, the number of good apartments could be minimum possible if, for example, apartments with indices 1, 2 and 3 were inhabited. In this case only apartment 4 is good. The maximum possible number could be, for example, if apartments with indices 1, 3 and 5 were inhabited. In this case all other apartments: 2, 4 and 6 are good.

C. Planning

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

Helen works in Metropolis airport. She is responsible for creating a departure schedule. There are n flights that must depart today, the i-th of them is planned to depart at the i-th minute of the day.

Metropolis airport is the main transport hub of Metropolia, so it is difficult to keep the schedule intact. This is exactly the case today: because of technical issues, no flights were able to depart during the first k minutes of the day, so now the new departure schedule must be created.

All n scheduled flights must now depart at different minutes between $(k+1)$-th and $(k+n)$-th, inclusive. However, it's not mandatory for the flights to depart in the same order they were initially scheduled to do so — their order in the new schedule can be different. There is only one restriction: no flight is allowed to depart earlier than it was supposed to depart in the initial schedule.

Helen knows that each minute of delay of the i-th flight costs airport c_i burles. Help her find the order for flights to depart in the new schedule that minimizes the total cost for the airport.

Input

The first line contains two integers n and k ($1 \le k \le n \le 300\,000$), here n is the number of flights, and k is the number of minutes in the beginning of the day that the flights did not depart.

The second line contains n integers $c_1, c_2, \ldots, c_n$ ($1 \le c_i \le 10^7$), here c_i is the cost of delaying the i-th flight for one minute.

Output

The first line must contain the minimum possible total cost of delaying the flights.

The second line must contain n different integers $t_1, t_2, \ldots, t_n$ ($k+1 \le t_i \le k+n$), here t_i is the minute when the i-th flight must depart. If there are several optimal schedules, print any of them.

Example

input

```
5 2
4 2 1 10 2
```

output

```
20
3 6 7 4 5
```

Note

Let us consider sample test. If Helen just moves all flights 2 minutes later preserving the order, the total cost of delaying the flights would be $(3-1)\cdot 4+(4-2)\cdot 2+(5-3)\cdot 1+(6-4)\cdot 10+(7-5)\cdot 2=38$ burles.

However, the better schedule is shown in the sample answer, its cost is $(3-1)\cdot 4+(6-2)\cdot 2+(7-3)\cdot 1+(4-4)\cdot 10+(5-5)\cdot 2=20$ burles.

D. Jury Meeting

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

Country of Metropolia is holding Olympiad of Metrpolises soon. It mean that all jury members of the olympiad should meet together in Metropolis (the capital of the country) for the problem preparation process.

There are $n+1$ cities consecutively numbered from 0 to n. City 0 is Metropolis that is the meeting point for all jury members. For each city from 1 to n there is exactly one jury member living there. Olympiad preparation is a long and demanding process that requires k days of work. For all of these k days each of the n jury members should be present in Metropolis to be able to work on problems.

You know the flight schedule in the country (jury members consider themselves important enough to only use flights for transportation). All flights in Metropolia are either going to Metropolis or out of Metropolis. There are no night flights in Metropolia, or in the other words, plane always takes off at the same day it arrives. On his arrival day and departure day jury member is not able to discuss the olympiad. All flights in Megapolia depart and arrive at the same day.

Gather everybody for k days in the capital is a hard objective, doing that while spending the minimum possible money is even harder. Nevertheless, your task is to arrange the cheapest way to bring all of the jury members to Metrpolis, so that they can work together for k days and then send them back to their home cities. Cost of the arrangement is defined as a total cost of tickets for all used flights. It is allowed for jury member to stay in Metropolis for more than k days.

Input

The first line of input contains three integers n, m and k ($1 \le n \le 10^5$, $0 \le m \le 10^5$, $1 \le k \le 10^6$).

The i-th of the following m lines contains the description of the i-th flight defined by four integers d_i, f_i, t_i and c_i ($1 \le d_i \le 10^6$, $0 \le f_i \le n$, $0 \le t_i \le n$, $1 \le c_i \le 10^6$, exactly one of f_i and t_i equals zero), the day of departure (and arrival), the departure city, the arrival city and the ticket cost.

Output

Output the only integer that is the minimum cost of gathering all jury members in city 0 for k days and then sending them back to their home cities.

If it is impossible to gather everybody in Metropolis for k days and then send them back to their home cities, output "`-1`" (without the quotes).

Examples

input
```
2 6 5
1 1 0 5000
3 2 0 5500
2 2 0 6000
15 0 2 9000
9 0 1 7000
8 0 2 6500
```
output
```
24500
```

input
```
2 4 5
1 2 0 5000
2 1 0 4500
2 1 0 3000
8 0 1 6000
```
output
```
-1
```

Note

The optimal way to gather everybody in Metropolis in the first sample test is to use flights that take place on days 1, 2, 8 and 9. The only alternative option is to send jury member from second city back home on day 15, that would cost 2500 more.

In the second sample it is impossible to send jury member from city 2 back home from Metropolis.

E. Boredom

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Ilya is sitting in a waiting area of Metropolis airport and is bored of looking at time table that shows again and again that his plane is delayed. So he took out a sheet of paper and decided to solve some problems.

First Ilya has drawn a grid of size $n \times n$ and marked n squares on it, such that no two marked squares share the same row or the same column. He calls a rectangle on a grid with sides parallel to grid sides *beautiful* if exactly two of its corner squares are marked. There are exactly $n \cdot (n - 1) / 2$ beautiful rectangles.

Ilya has chosen q query rectangles on a grid with sides parallel to grid sides (not necessarily beautiful ones), and for each of those rectangles he wants to find its *beauty degree*. Beauty degree of a rectangle is the number of beautiful rectangles that share at least one square with the given one.

Now Ilya thinks that he might not have enough time to solve the problem till the departure of his flight. You are given the description of marked cells and the query rectangles, help Ilya find the beauty degree of each of the query rectangles.

Input

The first line of input contains two integers n and q ($2 \le n \le 200\,000$, $1 \le q \le 200\,000$) — the size of the grid and the number of query rectangles.

The second line contains n integers $p_1, p_2, \ldots, p_n$, separated by spaces ($1 \le p_i \le n$, all p_i are different), they specify grid squares marked by Ilya: in column i he has marked a square at row p_i, rows are numbered from 1 to n, bottom to top, columns are numbered from 1 to n, left to right.

The following q lines describe query rectangles. Each rectangle is described by four integers: l, d, r, u ($1 \le l \le r \le n$, $1 \le d \le u \le n$), here l and r are the leftmost and the rightmost columns of the rectangle, d and u the bottommost and the topmost rows of the rectangle.

Output

For each query rectangle output its beauty degree on a separate line.

Examples

input

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

output

```
1
1
1
```

input

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

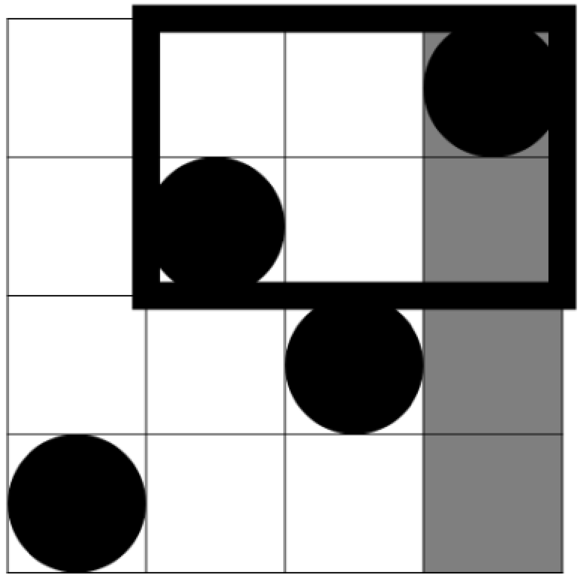
output

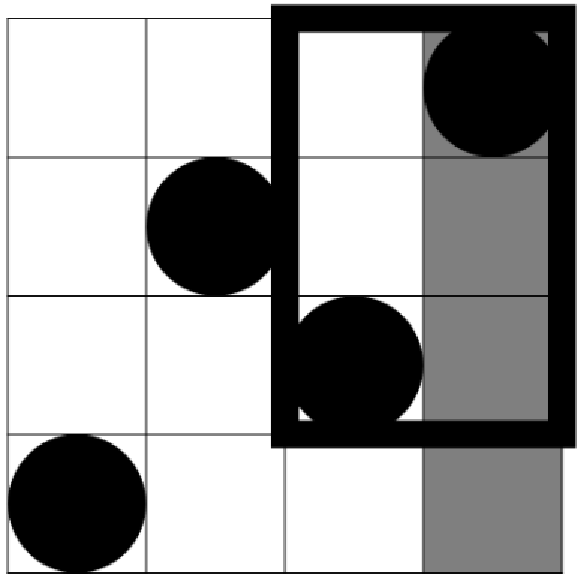
```
3
5
```

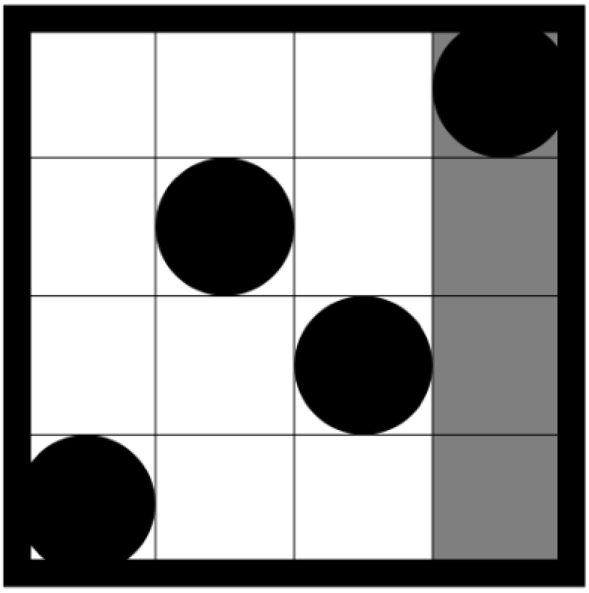
Note

The first sample test has one beautiful rectangle that occupies the whole grid, therefore the answer to any query is 1.

In the second sample test the first query rectangle intersects 3 beautiful rectangles, as shown on the picture below:







There are 5 beautiful rectangles that intersect the second query rectangle, as shown on the following picture: