2015 Canadian Computing Olympiad
Day 1, Problem 1
Hungry Fox

Time Limit: 2 seconds

Problem Description

It's dinner time for your pet fox! His meal consists of N crackers, with the ith cracker having a temperature of T_i degrees Celsius. He also has a large dish of water, which has a temperature of W degrees Celsius.

After taking an initial sip of water, your fox begins his meal. Every time he eats a cracker, its tastiness is equal to the absolute difference between its temperature, and the temperature of the last thing he ate or drank (be it the previous cracker he ate, or a sip of water, whichever he consumed most recently). He can drink some water whenever he wants, and can eat the crackers in any order.

Depending on the order in which your fox eats and drinks, the total tastiness of the crackers consumed may vary. What are the minimum and maximum values it can have?

Input Specification

The first line contains two integers, N ($1 \leq N \leq 100\,000$) and W ($0 \leq W \leq 10^9$), representing the number of crackers and the water's temperature. On the next N lines, there is one integer, T_i ($0 \leq T_i \leq 10^9$ for $1 \leq i \leq N$), representing the temperature of the ith cracker.

For at least 30% of the marks for this problem, $W = 0$.

Output Specification

The output is one line containing two integers: the minimum and maximum total tastiness your fox can experience during his meal, respectively.

Sample Input

```
3 20
18
25
18
```

Output for Sample Input

```
7 16
```

Explanation of Output for Sample Input

To minimize the total tastiness, the fox might drink water, eat the first cracker, eat the third cracker, drink more water, and finally eat the second cracker. He will then experience temperatures of 20, 18, 18, 20, and 25 degrees Celsius, and the crackers will have tastiness values of $2 + 0 + 5 = 7$.

To maximize the total tastiness, the fox might drink water, and then eat the crackers in order. He will then experience temperatures of 20, 18, 25, and 18 degrees Celsius, and the crackers will have tastiness values of $2 + 7 + 7 = 16$.

2015 Canadian Computing Olympiad
Day 1, Problem 2
Artskjid

Time Limit: 2 seconds

Problem Description

There are many well-known algorithms for finding the shortest route from one location to another. People have GPS devices in their cars and in their phones to show them the fastest way to get where they want to go. While on vacation, however, Troy likes to travel slowly. He would like to take the *longest* route to his destination so that he can visit many new and interesting places along the way.

As such, a valid route consists of a sequence of distinct cities $c_1, c_2, ..., c_k$, such that there is a road from c_i to c_{i+1} for each $1 \leq i < k$.

He does not want to visit any city more than once. Can you help him find the longest route?

Input Specification

The first line of input contains two integers n, m, the total number of cities and the number of roads connecting the cities ($2 \leq n \leq 18; 1 \leq m \leq n^2 - n$). There is at most one road from any given city to any other given city. Cities are numbered from 0 to $n - 1$, with 0 being Troy's starting city, and $n - 1$ being his destination.

The next m lines of input each contain three integers s, d, l. Each triple indicates that there is a one way road from city s to city d of length l km ($0 \leq s \leq n - 1; 0 \leq d \leq n - 1$; $s \neq d$; $1 \leq l \leq 10000$). Each road is one-way: it can only be taken from s to d, not vice versa. There is always at least one route from city 0 to city $n - 1$.

For at least 30% of the marks for this problem, $n \leq 8$.

Output Specification

Output a single integer, the length of the longest route that starts in city 0, ends in city $n - 1$, and does not visit any city more than once. The length is the sum of the lengths of the roads taken along the route.

Sample Input

```
3 3
0 2 5
0 1 4
1 2 3
```

Output for Sample Input

```
7
```

Explanation of Output for Sample Input

The shortest route would be to take the road directly from 0 to 2, of length 5 km. The route going from 0 to 1 to 2 is $4 + 3 = 7$ km, which is longer.

2015 Canadian Computing Olympiad
Day 1, Problem 3
Solar Flight

Time Limit: 15 seconds

Problem Description
A new era of aviation is upon us - the first solar-powered jumbo jets are about to be made available for public travel! However, this cutting-edge technology raises some safety concerns, as the rays of sunlight which power these planes can be blocked by other objects in the sky. As such, some statistics must first be calculated concerning the planned initial flights.

We consider a set of N flight paths, all travelling East from one city to another. A plane can be considered as a single point. The sky through which they pass can be modelled as a Cartesian plane, with x-coordinates representing distance East from an arbitrary fixed point, and y-coordinates representing altitude. We are interested only in the section of the sky with x-coordinates in the inclusive range $[0, X]$ ($1 \le X \le 10^9$), in which all flight paths are straight lines. The ith plane flies from (0, A_i) to (X, B_i) ($1 \le A_i, B_i \le 10^9$). All A values are distinct, as are all B values. The planes travel at unknown, possibly non-constant speeds along their flight paths, so at any point in time, a plane may be anywhere along its path. However, it is known that the planes will never crash with one another, so if two flight paths cross one another, both planes won't reach the intersection point at precisely the same time.

Planes also have an inteference factor: each plane i has an interference factor C_i ($1 \le C_i \le 10^9$), which is a measure of how much plane i would negatively impact the solar absorbing capability of any plane below them.

The solar panels on each plane are rather strange, in that they can only collect energy from directly above the plane. This means the sunlight that a given plane can absorb can be obstructed by other planes which occupy the same x-coordinate as it, and have a larger y-coordinate than it. In particular, their effectiveness is reduced based on the sum of the interference factor of all such planes.

Given this information, as well as a fixed distance constant K ($1 \le K \le X$), you must answer Q queries regarding the possible impact on planes' solar panels at various times. The ith query asks for the largest possible amount by which the plane P_i's solar panels could be obstructed at a single moment in time, at any point while the plane's x-coordinate is in the inclusive range $[S_i, S_i + K]$ ($0 \le S_i \le X - K$).

Input Specification
The first line contains four space-separated integers: X ($1 \le X \le 10^9$), the maximum x-coordinate to consider; K ($1 \le K \le X$), the fixed distance constant; N ($1 \le N \le 2000$), the number of flight paths; Q ($1 \le Q \le 800\,000$), the number of queries.

Each of the next N lines contain three integers, A_i, B_i, and C_i, for $i = 1..N$ ($1 \le A_i, B_i, C_i \le 10^9$), representing, for plane i, the starting y-coordinate, the ending y-coordinate, and the interference factor (respectively).

Each of the next Q lines contain two integers, P_i and S_i, for $i = 1..Q$ representing the query concerning plane P_i while its x-coordinate is in the range $[S_i, S_i + K]$.

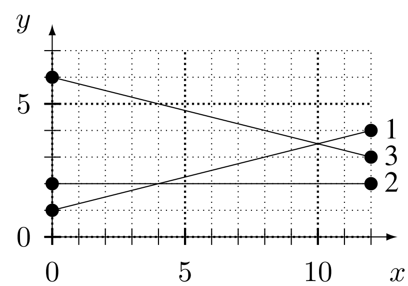
For 40% of the marks for this problem, $Q \leq 1000$.

Output Specification
The output consists of Q lines, each with the integer answer to the ith query, for $i = 1..Q$.

Sample Input
```
12 4 3 3
1 4 5
2 2 3
6 3 6
2 1
1 8
3 0
```

Output for Sample Input
```
11
6
0
```

Explanation of Output for Sample Input
Below is a diagram of the planes' flight paths:



The first query is for plane 2 over x-coordinates in the inclusive range $[1, 5]$. While this plane is at an x-coordinate smaller than or equal to 4, it might be covered by plane 3, but definitely not by plane 1. However, when it's at an x-coordinate larger than 4, it might be covered by both other planes. Therefore, the answer to this query is the sum of the other planes' interference factors, $5 + 6 = 11$.

For the second query, plane 1 could be covered by plane 3 while it has x-coordinate less than 10, and will not be covered by anything while it has x-coordinate greater than or equal to 10. Thus, it is only possibly interfered with by plane 3 with interference factor 6.

Neither plane 1 nor plane 2 can possibly be directly above plane 3 until it reaches x-coordinate 10, so the answer to the final query is 0.