C. Servers

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

There are n servers in a laboratory, each of them can perform tasks. Each server has a unique id — integer from 1 to n.

It is known that during the day q tasks will come, the i-th of them is characterized with three integers: t_i — the moment in seconds in which the task will come, k_i — the number of servers needed to perform it, and d_i — the time needed to perform this task in seconds. All t_i are distinct.

To perform the i-th task you need k_i servers which are unoccupied in the second t_i. After the servers begin to perform the task, each of them will be busy over the next d_i seconds. Thus, they will be busy in seconds $t_i, t_i + 1, ..., t_i + d_i - 1$. For performing the task, k_i servers with the smallest ids will be chosen from all the unoccupied servers. If in the second t_i there are not enough unoccupied servers, the task is ignored.

Write the program that determines which tasks will be performed and which will be ignored.

Input

The first line contains two positive integers n and q $(1 \le n \le 100, 1 \le q \le 10^5)$ — the number of servers and the number of tasks.

Next q lines contains three integers each, the i-th line contains integers t_i, k_i and d_i $(1 \le t_i \le 10^6, 1 \le k_i \le n, 1 \le d_i \le 1000)$ — the moment in seconds in which the i-th task will come, the number of servers needed to perform it, and the time needed to perform this task in seconds. The tasks are given in a chronological order and they will come in distinct seconds.

Output

Print q lines. If the i-th task will be performed by the servers, print in the i-th line the sum of servers' ids on which this task will be performed. Otherwise, print `-1`.

Examples

input

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

output

```
6
-1
10
```

input

```
3 2
3 2 3
5 1 2
```

output

```
3
3
```

input

```
8 6
1 3 20
4 2 1
6 5 5
10 1 1
15 3 6
21 8 8
```

output

```
6
9
30
-1
15
36
```

Note

In the first example in the second 1 the first task will come, it will be performed on the servers with ids 1, 2 and 3 (the sum of the ids equals 6) during two seconds. In the second 2 the second task will come, it will be ignored, because only the server 4 will be unoccupied at that second. In the second 3 the third task will come. By this time, servers with the ids 1, 2 and 3 will be unoccupied again, so the third task will be done on all the servers with the ids 1, 2, 3 and 4 (the sum of the ids is 10).

In the second example in the second 3 the first task will come, it will be performed on the servers with ids 1 and 2 (the sum of the ids is 3) during three seconds. In the second 5 the second task will come, it will be performed on the server 3, because the first two servers will be busy performing the first task.