1 Computing the Minimum Cost of a Flight

Problem Introduction

Now, you are interested in minimizing not the number of segments, but the total cost of a flight. For this you construct a weighted graph: the weight of an edge from one city to another one is the cost of the corresponding flight.

Problem Description

Task. Given an *directed* graph with positive edge weights and with n vertices and m edges as well as two vertices u and v, compute the weight of a shortest path between u and v (that is, the minimum total weight of a path from u to v).

Input Format. A graph is given in the standard format. The next line contains two vertices u and v.

Constraints. $1 \le n \le 10^4$, $0 \le m \le 10^5$, $u \ne v$, $1 \le u, v \le n$, edge weights are non-negative integers not exceeding 10^8.

Output Format. Output the minimum weight of a path from u to v, or -1 if there is no path.

Time Limits.

language	`C`	`C++`	`Java`	`Python`	`C#`	`Haskell`	`JavaScript`	`Ruby`	`Scala`
time (sec)	2	2	3	10	3	4	10	10	6

Memory Limit. 512MB.

Sample 1.

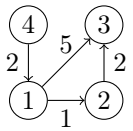
Input:

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

Output:

```
3
```

Explanation:



There is a unique shortest path from vertex 1 to vertex 3 in this graph ($1 \to 2 \to 3$), and it has weight 3.

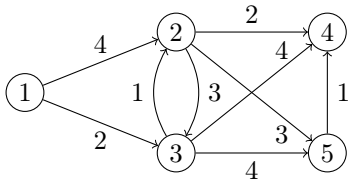
Sample 2.

Input:

```
5 9
1 2 4
1 3 2
2 3 2
3 2 1
2 4 2
3 5 4
5 4 1
2 5 3
3 4 4
1 5
```

Output:

```
6
```



There are two paths from 1 to 5 of total weight 6: $1 \to 3 \to 5$ and $1 \to 3 \to 2 \to 5$.

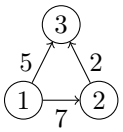
Sample 3.

Input:

```
3 3
1 2 7
1 3 5
2 3 2
3 2
```

Output:

```
-1
```



There is no path from 3 to 2.