

Problem A

Even Obsession

Patricia is an excellent software developer, but, as every brilliant person, she has some strange quirks. One of those is that everything she does has to be in even quantities. Most often that quirk does not affect her, even though it may seem strange to others. Some examples: every day she has to eat an even number of meals; during breakfast, she drinks two cups of coffee, eats two toasts and two slices of cheese; when she goes to the cinema she buys two tickets (fortunately she always has a friend that goes with her); she takes two baths per day (or four, or six...).

Some other times, however, that quirk makes the life of Patricia more difficult. For example, no one wants to travel by car with her because if she has to pay toll, the number of tolls she pays has to be an even number.

Patricia lives in a country where all roads are two-way and have exactly one toll each. She needs to visit a client in a different city, and wants to calculate the minimum total value of tolls she has to pay to go from her city to the client's city, obeying her strange quirk that she has to pay an even number of tolls.

Input

The input consists of several test cases. The first line of a test case contains two integers C and V , the total number of cities and the number of roads ($2 \leq C \leq 10^4$ and $0 \leq V \leq 50000$). The cities are identified by integer numbers from 1 to C . Each road links two different cities, and there is at most one road between each pair of cities. Each of the next V lines contains three integers C_1 , C_2 and G , indicating that the toll value of the road linking cities C_1 and C_2 is G ($1 \leq C_1, C_2 \leq C$ e $1 \leq G \leq 10^4$). Patricia is currently in city 1 and the client's city is C .

Output

For each test case in the input your program must output exactly one line, containing exactly one integer, the minimum toll value for Patricia to go from city 1 to city C , paying an even number of tolls, or, if that is not possible, the value -1 .

Examples

Input	Output
4 4 1 2 2 2 3 1 2 4 10 3 4 6 5 6 1 2 3 2 3 5 3 5 2 5 1 8 2 4 1 4 5 4	12 -1