Competitive Programming SS24

Submit until end of contest

Problem: beach (1.0 second timelimit)

Note: This is a problem that is harder to solve than usual. Solve the other problems first before spending too much time on this one.

This morning you made a list with chores that have to be done after the contest.

1. drop off letter
2. withdraw money
3. refuel car
4. buy groceries
5. bring them home
6. buy ice cream
7. visit nearest beach

Your outstanding time management abilities revealed, that the order is important. The ice melts if you buy it earlier, the letter is urgent, your car is nearly out of fuel etc. To chill at the beach as soon as possible, you decide to optimize in advance; during an event where you have plenty of free time: the competitive programming contest. Since you will be programming anyway, just write a small program that computes the shortest trip through the city (and nearby region) to get all the chores done in the correct order.

Input The first line contains the number of places n in your (connected) city and the number of streets m between them $2 \leq n, m \leq 10^5$. Then m lines follow for the streets with three integers describing the endpoints $0 \leq i, j < n$ and length $0 < w \leq 1000$. This means there is an undirected route from place i to place j that takes w minutes to travel by car.

The next line contains six integers $c_1, c_2, c_3, c_4, c_6, c_7$. The value of $1 \leq c_i \leq 100$ is the number of places where you can execute your i-th item on the list.

Then, six lines follow with c_i places each. The first of the six lines contains all the places where you can drop off a letter, the second line contains all the palaces where you can withdraw money and so on. You started numbering places when you were little, so your home is place 0 and the HPI is place 1. Needless to say, you start immediately after the contest from the HPI.

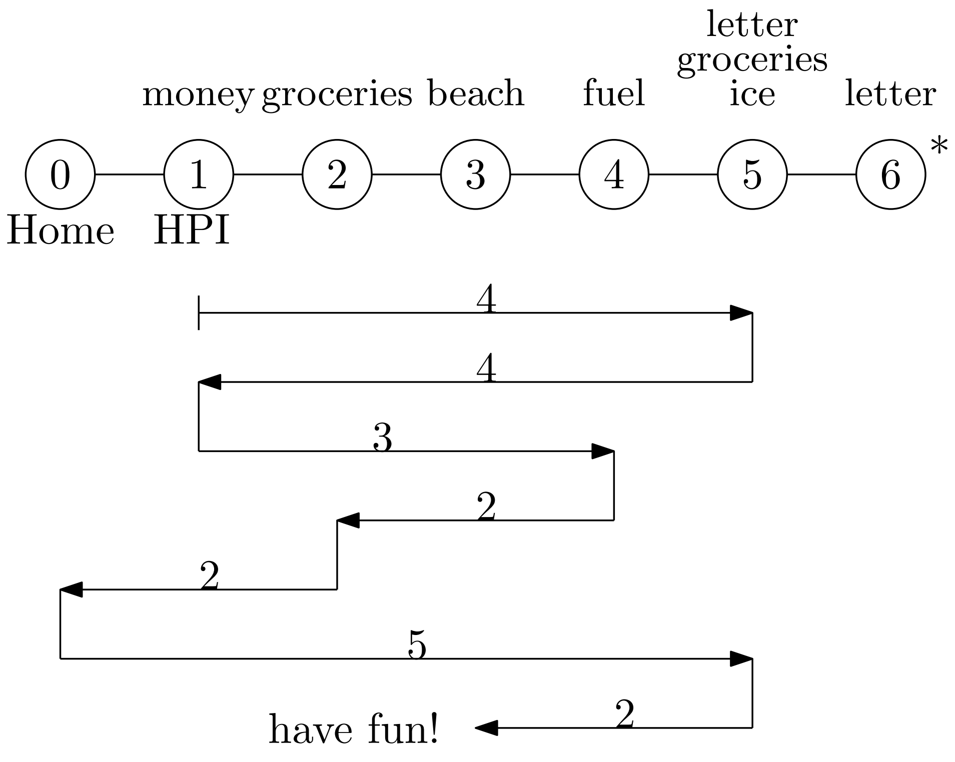
Output Print a line with the minimum travel time this afternoon.

Sample input

```
7 6
0 1 1
1 2 1
2 3 1
3 4 1
4 5 1
5 6 1
2 1 1 2 1 1
5 6
1
4
2 5
5
3
```

Sample output

```
22
```



*Input is not always a simple path