Homework Assignment #10: Programming Exercise #2

Due Date & Time

2:10PM Tuesday, April 14, 2016. Late submission will be penalized by 20% for each working day overdue.

Problem Description

Solve Problem C "Catering" of the 2015 Annual ACM International Collegiate Programming Contest World Finals (see the appended) or an alternative version with the following modifications:

1. The number of requests is never more than the number of catering teams.
2. Moving several catering teams (along with their equipments) simultaneously incurs the same cost as that of moving one team from one site to another.
3. Requests may be served out of time order and the cost of moving from site A to site B is the same as that of moving from site B to site A (i.e., the "edges" are undirected).

Notes

This assignment constitutes 4% of your grade. You may discuss the problem with others, but copying code is strictly forbidden. **Some of you may be requested to demonstrate your program.**

Submission Guidelines

- Indicate in a comment of your program which problem the program solves.
- Pack everything, excluding compiler-generated files, in a .zip file, named with the pattern "`b037050xx-alg2016-hw10.zip`".
- Upload the .zip file to the Ceiba course site for Algorithms 2016:

  `https://ceiba.ntu.edu.tw/1042algorithms`.
- If you use a Makefile, make sure that it outputs "`hw10`". Otherwise, make sure that the whole application can be compiled by a single command like "`gcc hw10.c`", "`g++ hw10.cpp`", or "`javac hw10.java`".

Grading

Your work will be graded according to its correctness and presentation. Before submission, you should have tested your program on several input cases. You should organize and document your program in such a way that other programmers, for example your classmates, can understand it. In the documentation, you may also want to explain how you have applied the algorithmic techniques, particularly design by induction and reduction, learned in class.

Below is a more specific grading policy:

Criteria	Score
incomplete or doesn't compile	≤ 20
complete, compiles, but with major errors	≤ 40
correct but with little documentation	≤ 80
correct and with good documentation	≤ 100
explanation of algorithmic techniques applied	$+10$

Problem C
Catering
Time limit: 4 seconds

Picture from Wikimedia Commons

Paul owns a catering company and business is booming. The company has k catering teams, each in charge of one set of catering equipment. Every week, the company accepts n catering requests for various events. For every request, they send a catering team with their equipment to the event location. The team delivers the food, sets up the equipment, and instructs the host on how to use the equipment and serve the food. After the event, the host is responsible for returning the equipment back to Paul's company.

Unfortunately, in some weeks the number of catering teams is less than the number of requests, so some teams may have to be used for more than one event. In these cases, the company cannot wait for the host to return the equipment and must keep the team on-site to move the equipment to another location. The company has an accurate estimate of the cost to move a set of equipment from any location to any other location. Given these costs, Paul wants to prepare an Advance Catering Map to service the requests while minimizing the total moving cost of equipment (including the cost of the first move), even if that means not using all the available teams. Paul needs your help to write a program to accomplish this task. The requests are sorted in ascending order of their event times and they are chosen in such a way that for any $i < j$, there is enough time to transport the equipment used in the i^{th} request to the location of the j^{th} request.

Input

The first line of input contains two integers n ($1 \leq n \leq 100$) and k ($1 \leq k \leq 100$) which are the number of requests and the number of catering teams, respectively. Following that are n lines, where the i^{th} line contains $n - i + 1$ integers between 0 and $1\,000\,000$ inclusive. The j^{th} number in the i^{th} line is the cost of moving a set of equipment from location i to location $i + j$. The company is at location 1 and the n requests are at locations 2 to $n + 1$.

Output

Display the minimum moving cost to service all requests. (This amount does not include the cost of moving the equipment back to the catering company.)

Sample Input 1	Sample Output 1

3 2
40 30 40
50 10
50
	80

Sample Input 2

```
3 2
10 10 10
20 21
21
```

Sample Output 2

```
40
```