Préparation au concours ACM
Année 2008-2009

B. Depardon, I. Morel, V. Sonigo

Partiel

"It will be found, in fact, that the ingenious are always fanciful, and the truly imaginative never otherwise than analytic."

The Murders in the Rue Morgue, E.A. Poe

1 Quelques directives

- Vous n'avez droit qu'à un seul ordinateur par groupe.
- Vous n'avez droit qu'aux pages sur la STL où les librairies standards, vous ne pouvez pas aller sur des forums ou toute autre page pouvant vous guider sur la solution.
- Vous n'avez pas droit à la page UVA toolkit (`http://uvatoolkit.com/problemssolve.php`), à vous de générer vos cas de test et de vérifier vos solutions.
- Tout document papier est autorisé dans une limite de 25 pages (police minimum 10), ainsi qu'un dictionnaire.
- Vous n'avez pas le droit de réutiliser vos anciens codes, ni même d'aller les relire (sauf s'ils sont dans votre document papier).
- Il y a 6 problèmes, vous avez 3 heures pour en traiter un maximum.
- À la fin de la séance, vous devez envoyer par mail aux 3 TD-{wo,}men vos codes dans un tarball ayant le nom suivant : `<login groupe>.tgz`. Chaque problème devra être dans un fichier ayant pour nom le numéro du problème (l'extension dépend bien entendu du langage utilisé).
- Petit rappel : traitez en priorité les exercices faciles.

2 The Settlers of Catan (539)

Within Settlers of Catan, the 1995 German game of the year, players attempt to dominate an island by building roads, settlements and cities across its uncharted wilderness.

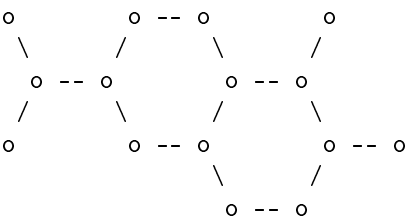
You are employed by a software company that just has decided to develop a computer version of this game, and you are chosen to implement one of the game's special rules:

When the game ends, the player who built the longest road gains two extra victory points.

The problem here is that the players usually build complex road networks and not just one linear path. Therefore, determining the longest road is not trivial (although human players usually see it immediately).

Compared to the original game, we will solve a simplified problem here: You are given a set of nodes (cities) and a set of edges (road segments) of length 1 connecting the nodes. The longest road is defined as the longest path within the network that doesn't use an edge twice. Nodes may be visited more than once, though.

Example: The following network contains a road of length 12.



2.1 Input

The input file will contain one or more test cases.

The first line of each test case contains two integers: the number of nodes n ($2 \leq n \leq 25$) and the number of edges m ($1 \leq m \leq 25$). The next m lines describe the m edges. Each edge is given by the numbers of the two nodes connected by it. Nodes are numbered from 0 to $n-1$. Edges are undirected. Nodes have degrees of three or less. The network is not neccessarily connected.

Input will be terminated by two values of 0 for n and m.

2.2 Output

For each test case, print the length of the longest road on a single line.

2.3 Sample Input

```
3 2
0 1
1 2
15 16
0 2
1 2
2 3
3 4
3 5
4 6
5 7
6 8
7 8
7 9
8 10
9 11
10 12
11 12
10 13
12 14
0 0
```

2.4 Sample Output

```
2
12
```

3 Always on the run (590)

Time limit: 3.000 seconds.

Screeching tires. Searching lights. Wailing sirens. Police cars everywhere. Trisha Quickfinger did it again! Stealing the 'Mona Lisa' had been more difficult than planned, but being the world's best art thief means expecting the unexpected. So here she is, the wrapped frame tucked firmly under her arm, running to catch the northbound metro to Charles-de-Gaulle airport.

But even more important than actually stealing the painting is to shake off the police that will soon be following her. Trisha's plan is simple: for several days she will be flying from one city to another, making one flight per day. When she is reasonably sure that the police has lost her trail, she will fly to Atlanta and meet her 'customer' (known only as Mr. P.) to deliver the painting.

Her plan is complicated by the fact that nowadays, even when you are stealing expensive art, you have to watch your spending budget. Trisha therefore wants to spend the least money possible on her escape flights. This is not easy, since airlines prices and flight availability vary from day to day. The

price and availability of an airline connection depends on the two cities involved and the day of travel. Every pair of cities has a 'flight schedule' which repeats every few days. The length of the period may be different for each pair of cities and for each direction.

Although Trisha is a good at stealing paintings, she easily gets confused when booking airline flights. This is where you come in.

3.1 Input

The input file contains the descriptions of several scenarios in which Trisha tries to escape. Every description starts with a line containing two integers n and k. n is the number of cities through which Trisha's escape may take her, and k is the number of flights she will take. The cities are numbered $1, 2, \ldots, n$, where 1 is Paris, her starting point, and n is Atlanta, her final destination. The numbers will satisfy $2 \leq n \leq 10$ and $1 \leq k \leq 1000$.

Next you are given $n(n-1)$ flight schedules, one per line, describing the connection between every possible pair of cities. The first $n-1$ flight schedules correspond to the flights from city 1 to all other cities $(2, 3, 4, \ldots, n)$, the next $n-1$ lines to those from city 2 to all others $(1, 3, 4, \ldots, n)$, and so on.

The description of the flight schedule itself starts with an integer d, the length of the period in days, with $1 \leq d \leq 30$. Following this are d non-negative integers, representing the cost of the flight between the two cities on days $1, 2, \ldots, d$. A cost of 0 means that there is no flight between the two cities on that day.

So, for example, the flight schedule "3 75 0 80" means that on the first day the flight costs 75, on the second day there is no flight, on the third day it costs 80, and then the cycle repeats: on the fourth day the flight costs 75, there is no flight on the fifth day, *etc.*

The input is terminated by a scenario having $n = k = 0$.

3.2 Output

For each scenario in the input, first output the number of the scenario, as shown in the sample output. If it is possible for Trisha to travel k days, starting in city 1, each day flying to a different city than the day before, and finally (after k days) arriving in city n, then print "The best flight costs x.", where x is the least amount that the k flights can cost.

If it is not possible to travel in such a way, print "No flight possible.".

Print a blank line after each scenario.

3.3 Sample Input

```
3 6
2 130 150
3 75 0 80
7 120 110 0 100 110 120 0
4 60 70 60 50
3 0 135 140
2 70 80
2 3
2 0 70
1 80
0 0
```

3.4 Sample Output

```
Scenario #1
The best flight costs 460.

Scenario #2
No flight possible.
```

4 Pascal's Triangle of Death (485)

Time limit: 3.000 seconds.

In this problem, you are asked to generate Pascal's Triangle. Pascal's Triangle is useful in many areas from probability to polynomials to programming contests. It is a triangle of integers with "1" on top and down the sides. Any number in the interior equals the sum of the two numbers above it. For example, here are the first 5 rows of the triangle.

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

In "Pascal's Triangle of Death," you are to generate a left justified Pascal's Triangle. When any number in the triangle exceeds or equals 10^{60}, your program should finish printing the current row and exit. The output should have each row of the triangle on a separate line with one space between each element.

The final element of each line should be directly followed by a newline. There is no space after the last number on each line.

4.1 Sample Input

There is no input for this problem.

4.2 Sample Output

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
.
.
.
```

etc.

5 Heavy Cargo (544)

Time limit: 3.000 seconds.

Big Johnsson Trucks Inc. is a company specialized in manufacturing big trucks. Their latest model, the *Godzilla V12*, is so big that the amount of cargo you can transport with it is never limited by the truck itself. It is only limited by the weight restrictions that apply for the roads along the path you want to drive.

Given start and destination city, your job is to determine the maximum load of the *Godzilla V12* so that there still exists a path between the two specified cities.

5.1 Input

The input file will contain one or more test cases. The first line of each test case will contain two integers: the number of cities n ($2 \leq n \leq 200$) and the number of road segments r ($1 \leq r \leq 19900$) making up the street network.

Then r lines will follow, each one describing one road segment by naming the two cities connected by the segment and giving the weight limit for trucks that use this segment. Names are not longer than 30 characters and do not contain white-space characters. Weight limits are integers in the range 0 - 10000. Roads can always be travelled in both directions.

The last line of the test case contains two city names: start and destination.

Input will be terminated by two values of 0 for n and r.

5.2 Output

For each test case, print three lines:

- a line saying "`Scenario #x`" where `x` is the number of the test case
- a line saying "`y tons`" where `y` is the maximum possible load
- a blank line

5.3 Sample Input

```
4 3
Karlsruhe Stuttgart 100
Stuttgart Ulm 80
Ulm Muenchen 120
Karlsruhe Muenchen
5 5
Karlsruhe Stuttgart 100
Stuttgart Ulm 80
Ulm Muenchen 120
Karlsruhe Hamburg 220
Hamburg Muenchen 170
Muenchen Karlsruhe
0 0
```

5.4 Sample Output

```
Scenario #1
80 tons

Scenario #2
170 tons
```

6 Longest Match (10100)

Time limit: 3.000 seconds.

A newly opened detective agency is struggling with their limited intelligence to find out a secret information passing technique among its detectives. Since they are new in this profession, they know well that their messages will easily be trapped and hence modified by other groups. They want to guess the intensions of other groups by checking the changed sections of messages. First they have to get the length of longest match. You are going to help them.

6.1 Input

The input file may contain multiple test cases. Each case will contain two successive lines of string. Blank lines and non-letter printable punctuation characters may appear. Each Line of string will be no longer than 1000 characters. Length of each word will be less than 20 characters.

6.2 Output

For each case of input, you have to output a line starting with the case number right justified in a field width of two, followed by the longest match as shown in the sample output. In case of at least one blank line for each input output "Blank!". Consider the non-letter punctuation characters as white-spaces.

6.3 Sample Input

```
This is a test.
test
Hello!

The document provides late-breaking information
late breaking.
```

6.4 Sample Output

```
 1. Length of longest match: 1
 2. Blank!
 3. Length of longest match: 2
```

7 Ecological Premium (10300)

Time limit: 1.000 seconds.
Memory limit: 32 MB.

German farmers are given a premium depending on the conditions at their farmyard. Imagine the following simplified regulation: you know the size of each farmer's farmyard in square meters and the number of animals living at it. We won't make a difference between different animals, although this is far from reality. Moreover you have information about the degree the farmer uses environment-friendly equipment and practices, expressed in a single integer greater than zero. The amount of money a farmer receives can be calculated from these parameters as follows. First you need the space a single animal occupies at an average. This value (in square meters) is then multiplied by the parameter that stands for the farmer's environment-friendliness, resulting in the premium a farmer is paid per animal he owns. To compute the final premium of a farmer just multiply this premium per animal with the number of animals the farmer owns.

7.1 Input

The first line of input contains a single positive integer n (< 20), the number of test cases. Each test case starts with a line containing a single integer f ($0 < f < 20$), the number of farmers in the test case. This line is followed by one line per farmer containing three positive integers each: the size of the farmyard in square meters, the number of animals he owns and the integer value that expresses the farmer's environment-friendliness. Input is terminated by end of file. No integer in the input is greater than 100000 or less than 0.

7.2 Output

For each test case output one line containing a single integer that holds the summed burden for Germany's budget, which will always be a whole number. Do not output any blank lines.

7.3 Sample Input

```
3
5
1 1 1
2 2 2
3 3 3
2 3 4
8 9 2
3
9 1 8
6 12 1
8 1 1
3
10 30 40
9 8 5
100 1000 70
```

7.4 Sample Output

```
38
86
7445
```