

X-CONTEST

Think... Create... Solve... & Have Fun!

**Faculty of Computer and Information Sciences
Ain Shams University
July 2011**

The problem set consists of 14 numbered pages

Problem [A]

Run-Length Encoding

Input file run.in

Code file run.cpp

Run-length encoding is a simple compression technique which compresses strings of letters by replacing repeated consecutive letters (called runs) by the number of occurrences of the letter, followed by that letter. For example, AAAABBBBCDDE compresses to 4A3BC2DE. The number 1 may be omitted in runs consisting of a single letter, as with letters 'C' and 'E' in the previous example.

Any string consisting of uppercase letters where each letter is optionally preceded by a positive integer is called a properly encoded string. Given a properly encoded string text, print out the decoded string. If the decoded string would be more than 50 characters long, print out "TOO LONG" (without the quotes).

Input Specification:

You'll be given a text which will contain between 0 and 50 characters ('0'-'9', 'A'-'Z'), inclusive. All numbers will be positive integers with no leading zeros and each number will precede a letter.

Note: Input ends at the end of file.

Output specification:

For each test case your program is required to output a line containing the decoded string.

Sample Input:

```
4A3BC2DE
1A1B1C1D1E
1A3A5A4BCCCC
50A
21Z13S9A8M
123456789012345678901234567890B
```

Sample Output:

```
AAAABBBBCDDE
ABCDE
AAAAAAAAABBBBCCCC
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
TOO LONG
TOO LONG
```

Problem [B]

Yummy, but...

Input file yummy.in

Code file yummy.cpp

Rafie (means Slim in English) is a food-fond guy. He loves eating so much that he recently gained a lot of weight. Rafie got sick and went to see a doctor. Finding Rafie's case very serious, the doctor advised him to organize and watch out for what he eats. He gave him a diet that he should follow, such that he shouldn't exceed a calories-limit per meal.

Rafie decided to follow the doctor's orders, but unfortunately Rafie visits his grandma every week-end - who is known for her delicious and yummy food. Rafie finds it very hard to resist his grandma delicious food and started to feel really bad.

Being Rafie's loyal friend, you decided to help him get over his problem. You decided to use your CS knowledge to make Rafie healthy and at the same time happy with eating his grandma's yummy food. Your program will help him eat the yummiest food and at the same time not exceed the calories-limit assigned by the doctor.

Input:

You trained Rafie to use your program such that he should input grandma's menu for the meal. The menu starts with the word "Menu" followed by the number of menu items then the menu items with their detailed information. Knowing that grandma is used to make a maximum of 30 menu items per meal, each menu item has a yummy-factor. The yummy-factor reflects how much this type of food is delicious & also how much Rafie loves it. Yummy-factor ranges from 1 (as Rafie loves all type of grandma's food) till 1000.

Menu items are listed by the item name, yummy-factor and the calories per each item; such that each food item calories ranges between 20 and 2000. If the menu item name contains a space, It will be input with a hyphen in the middle (ex: Fera7'-Mashweya).

Rafie should also input the doctor's calories-limit for the meal which by any means won't exceed 50,000.

Note: Input ends at the end of file.

Output:

Your program should output Rafie's healthy-meal. The output should be a list of items that Rafie should eat to be most satisfied (with the maximum yummy factor). In case more than two menu items can fit, then your program should output the one that appears first in the menu list.

In case your program failed to compute a yummy & healthy meal for Rafie, your program should output:
Sorry, Rafie. Couldn't find a healthy meal for you :(!

Note: Output an empty line after each test case.

Sample Input:

Menu
5
Fool 700 500
Ta3meya 300 800
Tamatem 50 900
Gebna-Beeda 70 400
3eesh 40 300
1200

Menu
2
Ma7shy 1000 900
Mesaba7a 700 50
500

Sample Output:

You are allowed to eat the following:

Fool
Gebna-Beeda
3eesh

You are allowed to eat the following:

Mesaba7a

Problem [C]

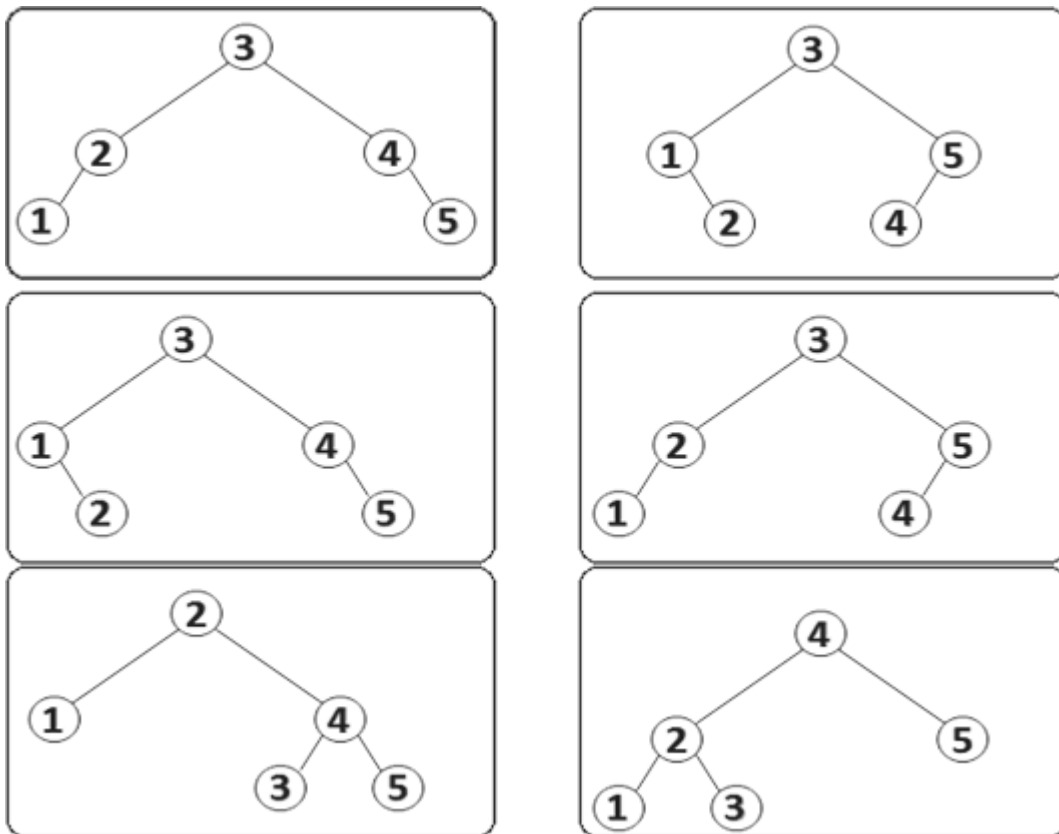
Semi-Complete Binary Search Tree!

Input file bst.in

Code file bst.cpp

- A **complete binary tree** is a binary tree in which every level, except possibly the last, is completely filled, and all nodes are as far left as possible.
- A **semi-complete binary tree** is a binary tree in which every level, except possibly the last, is completely filled. In other words, it is a complete binary tree without the second condition.
- A **semi-complete binary search tree** is a binary search tree that is semi-complete.

Having n distinct numbers, from 1 to n , there may be several ways to build a semi-complete binary search tree. For example, when $n = 5$, there are exactly **6 different ways to build a semi-complete binary search tree**.



Input:

Input consists of several test cases; each test case contains a non-negative number $n < 128$.

Output:

For each test case print n , adjusted in 3 places and the number of different ways a semi-complete binary search tree of n numbers can be built.

Sample Input:

```
0
1
2
5
53
91
```

Sample Output:

```
0 1
1 1
2 2
5 6
53 64512240
91 1118770292985239888
```

Problem [D]

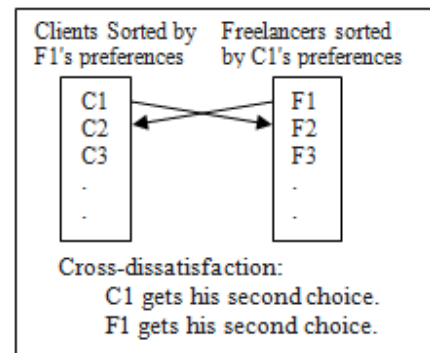
Broker

Input file broker.in

Code file broker.cpp

Aly is a broker (سمسار)! He assigns clients to *freelancers*. A freelancer is a software developer who works for himself, not for a company. No more than one client can be assigned to the same freelancer, and no more than one freelancer can be assigned to the same client. Aly wants to assign clients to freelancers in a smart way. So he gives each client the resumes of all freelancers, and asks each client to sort the freelancers according to the client's preferences. Aly also sorts the clients based on their reputations. The client with the highest reputation gets his first preference. The client with the second highest reputation gets his first preference if that first preference is not already assigned to the client with the highest reputation; otherwise the client with the second highest reputation gets his second preference. And so on.

However, Aly realizes that some freelancers abandon the clients that he assigns them to, and go for other clients who also abandon the freelancers that he assigns them to. Aly realizes that the reason his brokerage assignment sometimes fails is that he doesn't take into account the preferences of the freelancers themselves. For example, assume that the client C1 chooses the freelancer F1 to be his first preference, but because C1 has a low reputation, Aly assigns C1 to another freelancer F2 that was C1's second preference. Meanwhile, Aly assigns F1 to another client C2 who also has F1 as a first preference, but has better reputation than C1. If freelancers have no opinion, there will be no problem. But freelancers are, after all, free! They don't necessarily obey the broker. If F1 prefers to work for C1 rather than C2, then both C1 and F1 will abandon Aly's assignments and will go for each other. Aly calls this situation: "cross-dissatisfaction".



So Aly makes a small change in his policy. He asks each freelancer to sort the clients according to the freelancer's preferences. He still uses his old policy of assigning clients to freelancers based on client reputation and preferences, but only **after** he makes sure that there is no cross-dissatisfaction. In other words, he wants to find an assignment with no cross-dissatisfaction such that the assignment satisfies each client's preferences as much as possible, while giving higher priority to clients with higher reputation.

Input Specifications:

The input starts with a positive integer k , which indicates the number of test cases. Then k test cases follow. Each test case is organized as follows:

One line contains two space-separated integers: N and M . N is the number of clients, M is the number of freelancers, and $1 \leq N, M \leq 1000$.

[Each client has a unique ID from 1 to N , inclusive, and each freelancer has a unique ID from 1 to M , inclusive]

One line contains N integers, each represents the reputation level of a client, beginning from the client with ID=1, to the client with ID= N . Reputation levels range from 1 to N , with 1 being the highest reputation. No two clients have the same reputation level.

N lines, each contains the preferences of a client, beginning from the client with ID=1, to the client with ID= N . The preferences of a client are represented as a space-separated list of freelancer IDs sorted by the client's preferences. The first preference comes first.

M lines, each contains the preferences of a freelancer, beginning from the freelancer with ID=1, to the freelancer with ID=M. The preferences of a freelancer are represented as a space-separated list of client IDs sorted by the freelancer's preferences. The first preference comes first.

Output Specifications:

For each test case output the case number, followed by a hyphen (-), followed by Aly's brokerage assignment, all in a separate line. A brokerage assignment consists of N space-separated integers, each integer represents the ID of a freelancer that is assigned to a client, beginning from the client with ID=1, to the client with ID=N. If a client is not assigned a freelancer, put 0 instead of the freelancer's ID.

Sample input:

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

Sample output:

```
1-1 2 3
2-3 2 4 1
```


Problem [E]

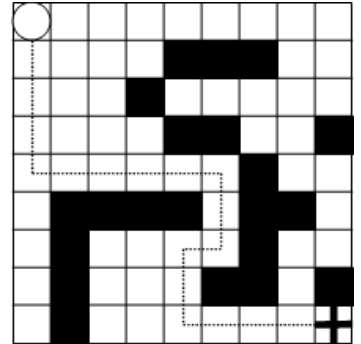
The robot (18-س)

Input file robot.in

Code file robot.cpp

(18-س) is an ancient robot that a team of researchers found in the Atlantic Ocean. You need to re-program (18-س) to make it capable of living in our modern world that is full of obstacles!

(18-س) will always move on a square grid of k meters length and k meters width. (18-س) can only move from one empty square to another, either horizontally or vertically, but not diagonally. There are objects that (18-س) need to avoid while moving. The edges of these objects are along the grid lines. Given the locations of these objects, your job is to compute the shortest path for (18-س) from its position at the upper left corner to the lower right corner.



Input specification:

The input starts with a line containing an integer N that is the number of test cases. Then N lines follow. Each line starts with an integer k that is the length (and width) of the square grid, then an integer M that is the number of objects on the grid, followed by M space-separated object locations. Each object location is written as two space-separated integers indicating the row and the column of the object. The upper-most row is row 0, and the left-most column is column 0. The dimensions of the grid are at most 100×100 .

Output specification:

For each test case output a line containing a single integer that is the length of the shortest path from the upper left square corner to the lower right corner. The length of the path is measured in squares, not including the source and the destination. You can assume that the destination is always reachable from the source.

Sample Input:

```
2
9 21 1 4 1 5 1 6 2 3 3 4 3 5 3 8 4 6 5 1 5 2 5 3 5 4 5 6 5 7 6 1 6 6 7 1 7 5 7 6 7 8 8 1
100 0
```

Sample Output:

```
17
197
```

Problem [F]

The Chicken and Egg

Input file chicken.in

Output file chicken.cpp

Abu-Said is a farmer from Kafr Abocomp. He raises chicken. He begins the season with a number of chickens. Each chicken produces one egg in the morning. Each day he goes to the market and sells all the eggs he got this day. He also brings along a few chickens and sells some of them.

In Kafr Abocomp market, where Abu-Said sells his chickens and eggs, the price of the eggs never changes during the season. However, the price of the chicken is different each day. Abu-Said is guaranteed to sell all the eggs produced that day but he is not guaranteed to sell all the chickens. He only gets an offer for buying a number of chickens a day and he is never able to sell one chicken more than what he was offered.

If Abu-Said sold any chicken during a day, the next day he will have fewer eggs to sell. Abu-Said doesn't let the eggs hatch and he doesn't get any chickens during the season. So, selling a chicken is not an easy decision as it means the sold chicken will not produce eggs for the rest of the season. This means he will sacrifice the money he will get from selling the eggs produced by the sold chicken for the rest of the season. So even if Abu-Said gets an offer for buying a number of chickens, he doesn't always sell the number offered. He can keep some chicken and sell some based on the price of chicken that day and whether it will be better to sell the chicken or keep it to produce eggs.

Abu-Said was able to get an estimation of the daily price of the chicken and the number of chickens that he will be offered to buy each day in the season. He wants your help in the decision of how many chickens he should sell daily in order to maximize his profit. Abu-Said can figure that out if you gave him the maximum total profit he can get in the season. Abu-Said's profit in the season is the sum of money he got from the chickens and eggs he sold in all days during the season.

Input Specification

The input file will consist of a number of seasons. Each season will begin with 3 numbers on a single line: d , c , and e where $2 \leq d, c, e \leq 1000$. d is the number of days in the season, c is the number of chickens Abu-Said will start his season with and e is the price of one egg in any day in the season.

The line will be followed by d lines; each represents the data of a single day in the season. The first line is data for the first day and the second is data of the second day and so on. Each day data is a pair of integers p and n where $0 \leq p \leq 10000$ and $0 \leq n \leq 10$, p is the price of one chicken on that day and n is the number of chicken Abu-Said will be offered to sell on that day. Each season will be followed a blank line. The input ends when no more seasons are given.

Output Specification

The output should be an integer n per season on a single line. Where n is the maximum profit Abu-Said can make from selling the chicken and eggs in all the days of the season. You can assume the chicken will live till the end of the season and that any chicken remaining after the season will never be sold.



Sample Input

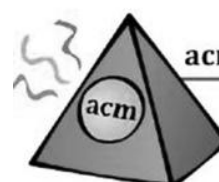
```
2 2 10
1 2
20 2
```

```
2 2 10
20 1
1 1
```

```
2 2 10
5 2
1 2
```

Sample Output

```
80
51
42
```



Problem [G]

Shalabya party

Input file shalabya.in

Output file shalabya.cpp

We all know Ra'afat El-Haggan and Gomaa El-Shawan our heroes, but not all of us know that their original names are "Refa'at El-Gammal" and "Ahmad El-Hawan" respectively. They were spies that worked for GIS (General Intelligence Services). The thing that we did not know is that in FCIS there is one hero like those. She is called "Shalabya", her colleagues love her, because she is very active person and always helpful. Shalabya was an intelligent 4th year student. It's important to state the fact that she never cared about the grades, her only devotion was to science and from day one in the college her motto was **"Science not Grades!"** that's why she got an Excellent GPA that she deserved.

Coincidentally, her birthday was on the last day of exams, so her friends wanted to make a special birthday, but as you know those heroes never publish any event related to them with their original names, but most likely aliases (أسماء مستعارة), because of that, they called the day "Galabya Party" instead of "Shalabya party". Because she is intelligent and prudent she told her friend "Naseefa" to make an event on Facebook and to make it "as" real event to invite 4th year students to come with their Galabyas. And all of 3rd and 4th year students came to this party. Naseefa made dish party in this day, but not like the ordinary ones, she is creative so it was like a contest between her friends to see the best in cooking between them.

On the announcement of the results of this contest, something dangerous happened; some important files had been stolen from College and Dr. Bayoumy (famous professor there) asked her to leave the party and help him to get the thief. Since has very a good memory, any person is represented in her head as pair of 2 numbers, ID and this person's year in the college respectively (this ID is square numbers for 4th year, and linear for 3rd year), Since all 3rd and 4th year students came to this party and only the thief has left, so it will be easy to find the thief if she subtract summation of all persons' IDs from total that she already knows. Unfortunately, she was very tired from the party especially from the food, so she asked you to help her to just get this sum. But since she knows that you have eaten with her so you will be lazy to sum all students, as a result of this, she created a new way to sum all student using just sample of them, by following approach:

For a positive integer n , let $f(n)$ denote the sum of the digits of n when represented in base 10. It is easy to see that the sequence of numbers $n, f(n), f(f(n))$ and $f(f(f(n)))...$ eventually becomes a single digit number that repeats forever. Let this single digit be denoted by $g(n)$.

- 1- You will be given a sequence (V) of randomly chosen N integers, these numbers indicating IDs for chosen students.
- 2- Multiply all these numbers $P := (\prod_{i=0}^N v[i]) \text{ MOD } 1,000,000,007$
- 3- Multiply $g(P) * P * N$ $M := (g(P) * P * N) \text{ MOD } 1,000,000,007$
- 4- The total result Shalabya wants is:

$$\left(6 * \sum_{i=0}^M i^2 + 2 * \sum_{i=0}^M i \right) \text{MOD } 1,000,000,007$$

For example, consider $n = 1234567892$. Then: $f(n) = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 2 = 47$

$f(f(n)) = 4 + 7 = 11$

$f(f(f(n))) = 1 + 1 = 2$, Therefore, $g(1234567892) = 2$.

Input Specification

You will be given N and then N numbers representing values on sequence V , N is a positive integer of maximum 50. Each element in sequence V is a positive integer with maximum value = 10,000

Output specification

You are required to print out the result Shalabya wants on a single line.

Sample Input

```
15 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
8 48 12 56 199 200 1020 45 2
```

Sample Output

```
1370105
392852425
```



Problem [H]

Beauty of Numbers

Input file numbers.in

Output file numbers.cpp

There is nothing more beautiful than just an integer number.

Input Specification:

You are given two integers n and k , where n will be between 1 and 10^{18} , inclusive and k will be between 1 and 10, inclusive.

Note: Input ends at the end of file.

Output Specification:

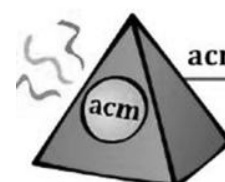
You're required to print the smallest integer greater than or equal to n that contains exactly k distinct digits in decimal notation.

Sample Input:

```
47 1
7 3
69 2
12364 3
```

Sample Output:

```
55
102
69
12411
```



Problem [I]

Palindrome**Input file** palindrome.in**Output file** palindrome.cpp

A numerical sequence is called a palindrome if the reverse of the sequence is the same as the original. For example sequences {1, 2, 1}, {15, 78, 78, 15} and {112} are palindromes, but {1, 2, 2}, {15, 78, 87, 51} and {112, 2, 11} are not.

Input Specification:

You will be given a sequence which will contain between 1 and 50 integers, inclusive and each element of the sequence will be between 1 and 10000, inclusive. You can replace any two adjacent numbers with their sum.

Note: Input ends at the end of file.

Output Specification:

Your program should output a minimal number of such operations required to make the given sequence a palindrome on a single line for each case.

Sample Input:

15 78 78 15

1 1 1 3

15 78 87 51

3 23 21 23 42 39 63 76 13 13 13 32 12 42 26

Sample Output:

0

2

3

8