The ACM International Collegiate Programming Contest
Sponsored by IBM

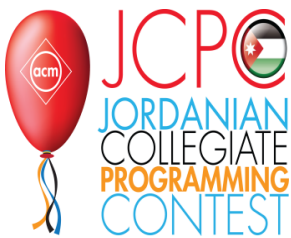International Collegiate
Programming Contest

IBM | event sponsor

The Second Jordanian Collegiate Programming Contest

ACM International Collegiate Programming Contest

ARAB COLLEGIATE
programming contest

Princess Sumaya University for Technology (PSUT)
Amman, Jordan
November 2012

Program:	fegla.(c|cpp|java)
Input:	fegla.in
Balloon Color:	red

[A] Fegla and the Bed Bugs

Fegla, also known as mmaw, is coaching a lot of teams. All these teams train together in one place, unfortunately this place doesn't have any good ventilation and is quite small relative to the number of teams. All these circumstances resulted in a strange creature appearing! That creature is called The Bed Bug!

These are parasitic bugs; they feed on human blood by biting them. What was strange and confused Fegla, is that some of the team members did not get bitten at all! However, he was more interested in eliminating these bugs. After observing the bugs' behavior for some time, he concluded that he needed to stop them from reproducing to eliminate them. They reproduce by getting very close to each other.

And so, Fegla needs your help. Given a straight line of empty cells N and the number of bugs K, tell Fegla the best assignment for the bugs to maximize the minimum number of empty cells between each two consecutive bugs on that line.

For example, given N=4 and K=2, the answer would be 2, according to the best assignment:

Bed Bug	Empty	Empty	Bed Bug

Input Specification

Input will start with an integer T representing the number of test cases. Followed by T lines each line contains two integers N, K.

You can assume that

$2 <= N <= 200$
$2 <= K <= N$

Output Specification

For each test case in a separate line, output the minimum distance between EACH two consecutive bugs in the best assignment.

Sample Input

```
2
4  2
3  2
```

Sample Output

```
2
1
```

Program:	nicole.(c|cpp|java)
Input:	nicole.in
Balloon Color:	gold

[B] Do Not Disturb!

Nicole and Noura are from the media team of the regional contest, Noura takes photos of the teams in the contest hall during the contest, meets up with Nicole, and then Nicole uploads them to the Facebook page of the contest. In order not to disturb the teams, the contest director sets some paths that Nicole and Noura can use while doing this task. These paths intersect with each other forming a graph.

Formally, you are given a graph of nodes representing the points of intersection and the end points of the paths. You are given also bidirectional edges connecting the nodes. Two nodes are considered neighbors if there is an edge connecting them. You can assume that it takes one time unit to pass any edge. At the start of the contest, Nicole and Noura are in two nodes (not necessarily different node). At each time unit Noura and Nicole will each choose a neighboring node to go to or choose to stay at their current nodes for a full time unit with each possibility having an equal probability of being chosen. Your task is to compute the expected number of time units before they meet for the first time in a predefined node (which has the machine that will be used for uploading photos).

Input Specification

The first line of the input starts with an integer T representing the number of test cases.
Each test case starts with a line containing two space separated integers V and E with V representing the number of nodes and E representing the number of edges.
The second line of each test case has three space separated integers A, B and C where A is the starting node of Nicole, B is the starting node of Noura and C is the index of the node where they will meet.
E lines follow, each having two space separated integers F and T representing an undirected edge between nodes F and T (note that there is at most one edge between any pair of nodes and there is no edge from a node to itself)

$1 \le V \le 20$
F, T, A, B, C are all zero based indices

Output Specification

For each test case print on a single line the expected number of moves rounded to three decimal places, if they can never reach each other print "Impossible" instead.

Sample Input

```
3
3 2
0 2 1
0 1
2 1
1 0
0 0 0
2 1
0 0 1
0 1
```

Sample Output

```
4.800
0.000
4.000
```

[C] Casino

Program:	dalia.(c|cpp|java)
Input:	dalia.in
Balloon Color:	pink

Dalia is the assistant director of the fundraising team in the ACPC. She is facing a difficult time this year, there's a huge lack of sponsors! And now we are facing the danger of not being able to provide the teams with balloons, T-shirts or even name-tags.

Dalia knows it is too late to get a sponsor, actually too late to do anything. But she doesn't simply give up; she decided that her only hope is to gamble. She will go to a casino where they just invented a new game; she thinks she might have a more promising chance if she plays that game.

The game is very simple, the dealer puts a long string of cards on the table in front of Dalia and she is required to point out palindromes longer than one character (words that are read backward the same as forward) of maximum length (a maximum length palindrome is a palindrome that no other palindrome exists in the string with greater length). So if the maximum length of palindrome in a string is $X>1$, print all palindromes of length X in the string.

Input Specification

Input will start with T number of test cases. Each test case will consist of 1 line that contains a non-empty string S of lower case English letters no longer than 1000 characters.

Output Specification

For each test case, print a line containing the case number as shown in the sample then print the palindromes each on a line by itself, in the order of their occurrence in S from right to left.

Sample Input

```
2
abcba
abba
```

Sample Output

```
Case #1:
abcba
Case #2:
abba
```

Program:	fouad.(c|cpp|java)
Input:	fouad.in
Balloon Color:	green

[D] Fundraised

Guess what?! Dalia won the money (an amount X), and now it's time to spend it. Mohamed Fouad is the deputy regional contest director and he wants to spend the money but he doesn't know what to spend it on. There are N items that he can buy like name-tags, T-shirts, helium balloons, trophies, etc. each having a different importance value and a cost to buy one unit from (you can buy as much units of any item as you want as long as you don't exceed the budget).

Mohamed wants your help; tell him the maximum importance value he can get after he chooses optimally to buy any number of items without exceeding his budget.

Input Specification

The first line of input contains an integer T, the number of test cases. T test cases follow, the first line of each test case contains two integers ($1 <= N <= 100$), ($1 <= X <= 10000$). The second line of the test case contains N space separated integers I_i ($1 <= I_i <= 400000$), each representing the importance value Fouad earns from item i ($0 <= i < N$). The following line contains N space separated integers C_i ($1 <= C_i <= 1000$), each representing the cost of an item i ($0 <= i < N$).

Output Specification

For each test case, on a separate line, output the maximum importance value Fouad can get.

Sample Input

```
1
5 20
1 2 3 4 5
2 6 3 5 4
```

Sample Output

```
25
```

Program:	meeda.(c|cpp|java)
Input:	meeda.in
Balloon Color:	blue

[E] You Shall Not Pass!!

Ahmed Aly is the problem architect for this year's regional contest, and he knows a lot about the participating teams and their coaches. More precisely, each participating university has a coach who coaches some seniors; each of them might be coaching some juniors where each junior might be coaching another junior, and so on. You can think of this as a hierarchy relation, where each team coaches those who are directly beneath it in the hierarchy. Ahmed, of course, knows these relations precisely and knows who coaches whom.

Ahmed also knows that, for each team, he can write a problem that he knows neither this team nor its trainees nor anyone beneath them in the hierarchy can solve and everyone else will solve it, but he has a limited number of problems he can put in the problem set of the contest. He wants to use all that for the sole purpose of maximizing the number of teams who will not solve at least one problem.

Ahmed needs your help, write a program that given the number of problems Ahmed can write in the problem set and the relations between the teams, prints the maximum number of those who will not solve at least one problem.

Note that a team can only be coached by one other team

Input Specification

The first line of input contains an integer *T*, the number of test cases. *T* test cases follow, the first line of each test case contains three integers *A*, *B*, *C*, the number of teams, the number of relations and the number of problems available in the problem set respectively. Then follows *B* lines, each in the format *F T*, where *F* and *T* are contestants and *F* coaches *T* where contestants are numbered from 0 to *A-1*.

$0 < T \leq 100$
$0 < A \leq 10000$
$0 \leq B < A$
$0 \leq C \leq A$
$0 \leq F, T < A$

Output Specification

For each test case, on a separate line, output the maximum number of those who will not solve at least one problem.

Sample Input

```
2
2 1 1
0 1
10 7 2
0 3
4 1
3 2
8 5
6 9
8 6
5 7
```

Sample Output

```
2
8
```

[F] It Is Cold

Program:	ziad.(c|cpp|java)
Input:	ziad.in
Balloon Color:	white

Dr. Ziad Najem is known as the godfather of the ACPC. When the regional contest was held in Alexandria, Egypt, the weather was very cold. What surprised Dr. Ziad was that in the contest hall the fans were turned on! Dr. Ziad immediately needed to know, for each team, the speed in which the air reached that team.

Each team has N fans placed on a straight line to its right. Each fan i has a speed S_i and direction C_i. Directions are either towards the team "T" or away from the team "A".

If two fans face the same direction, their speeds add up in the same direction. E.g.

	Fan 1	**Fan 2**	**Result**
Direction	T	T	T
Speed	2	4	2+4 = 6

If two fans face each other, their speeds cancel out. E.g.

	Fan 1	**Fan 2**	**Result**
Direction	A	T	T
Speed	2	4	4-2 = 2

Input Specification

The first line of input contains an integer T, the number of teams. For each team, you will be given an integer N, the number of fans. Two lines follow; the first line contains N space separated integers S_i, the speed of the air produced by fan i ($0 \le i < N$). The second line contains N characters that describe the direction of the fans C_i (as described above).

T <= 100
0 < N <= 1000
0 <= Si <= 1000000000

Output Specification

There should be T lines, containing a single integer S each that represents the speed in which the air reaches the team. S is zero if no air reaches the team.

Sample Input

```
3
2
1  2
T  T
2
3  5
A  T
2
3  5
T  A
```

Sample Output

```
3
2
3
```

Program:	sayed.(c|cpp|java)
Input:	sayed.in
Balloon Color:	black

[G] Passwords

Ahmed El-Sayed, the ACPC Regional System Administrator, has generated passwords for each team to login to the pc^2 program used to run the contest. Unfortunately for him, he lost these passwords! But fortunately for us, he remembered how he generated them.

To generate the passwords, Ahmed has a list X of N strings. He defines a list P as a permutation of list X if and only if P and X contain the same elements but not necessarily in the same order. He then generates all possible permutations of X. He then creates a new list Y such that it contains a string S for each permutation P of X, S is the concatenation of all the strings in the permutated list P. Finally, he sorts the list Y in lexicographic order (according to the ASCII table) and removes duplicate strings from Y. Then the passwords are the strings at some predefined indices in Y.

Given index K (starting from 1) and the initial list of strings X, you should tell Ahmed the password in index K.

Input Specification

The first line of input contains an integer T, the number of test cases. Each test case consists of two integers $N <= 20$ (size of list X) and K (always a valid value) followed by N strings. Each string consists of maximum 10 characters and all strings will be of equal size. The strings have no whitespaces and consist only of alphabetical characters.

Output Specification

For each test case, on a separate line, print the Kth password as explained above.

Sample Input

```
1
3 3
w
y
d
```

Sample Output

```
wdy
```

[H] Skill

Program:	yasser.(c|cpp|java)
Input:	yasser.in
Balloon Color:	silver

Yasser is an Egyptian coach; he will be organizing a training camp in Jordan. At the end of camp, Yasser was quiet amazed that the participants solved all of the hard problems he had prepared; so he decided to give them one last challenge:

Print the number of integers having *N* digits where the digits form a non decreasing sequence.

Input Specification

Input will start with **T** <= 100 number of test cases. Each test case consists of a single line having integer *N* where 1 <= *N* <= 100000.

Output Specification

For each test case print on a separate line the number of valid sequences modulo 1000000007.

Sample Input

```
3
2
3
4
```

Sample Output

```
55
220
715
```

[I] Eligibility

Regional Contest Director Osama Ismail knows that an individual cannot participate in the regional contest more than 5 times. Given a list of contestant names and the previous years they participated in, print which contestant is eligible.

Input Specification

The first line of the input contains a single integer T representing the number of the test cases
The first line of each test case contains a single integer N
N lines follow in this test case each having the format "Name Year" denoting that contestant Name participated in a regional contest in year Year

$T \leq 100$
$0 \leq N \leq 500$
Name is sequence of lowercase English letters, spaces and contains up to 20 characters
$1970 \leq Year \leq 2070$

Note that since he collected the data from multiple sources it may contain duplicate records (if a contestant X have competed in year Y, you might find multiple lines "X Y" in the same test case)

Output Specification

For each test case, print a line containing the test case number as formatted in the sample and then for each eligible contestant print his\her name on a single line and note that you must print the names of the contestants in lexicographic order

Sample Input

```
1
6
ahmed 2010
ahmed 2011
ahmed 2009
ahmed 2008
ahmed 2005
mohamed 2001
```

Sample Output

```
Case #1:
mohamed
```