2015 Taiwan National Collegiate Programming Contest

Preliminary Round: Oct. 4. 2015

- Problems: There are 6 problems (11 pages in all) in this packet.
- Program Input: Input to the program are through standard input. Each input file may contain one or more test cases. Test cases may be separated by any delimiter as specified in the problem statements.
- Program Output: All output should be directed to the standard output (screen output).
- Time Limit: The judges will run each submitted program with certain time limit (given in the table below).

Table 1: Problem Information

	Problem Name	Time Limit
Problem A	Sum of Largest Indices	1 sec.
Problem B	Gas Problem	1 sec.
Problem C	School Hunting	3 sec.
Problem D	Sum and Product of Eigenvalues	3 sec.
Problem E	Acquition Under Budget	1 sec.
Problem F	Noah 2	1 sec.

Problem A
Sum of Largest Indices
Time Limit: *1 Second*

Let $A[1]..A[n]$ be an array of n positive integers. For each distinct integer $x \in A$, let $L(x)$ denote the largest index of x in A. You are asked to compute $\sum_{x \in A} L(x)$.

Technical Specification

Note that n is less than or equal to 10^5. Any integer in A is less than or equal to 1000.

Input File Format

The first line of the input file contains an integer denoting the number of test cases. There are at most 10 test cases. In each test case, there is a line listing all the integers in A. Note that there is a blank between any two integers.

Output Format

For each test case, output its answer in one line.

Sample Input

```
3
3 1 2 4 3 5 1 4 4
3 1 2 3 3 5 1 4
1 1 1 3 3 1 3 9 9 1000
```

Output for the Sample Input

```
30
29
32
```

Problem B
Toll problem
Time Limit: *1 Second*

A popular attraction at the ACM theme park is the petting zoo. There are many free petting animal stations in the zoo. One must ride the in-zoo go-cart to go from station to station. To ride the go-cart, one must have enough zoo coins to travel between two stations. After petting the animal at a station, possibly more zoo coins may be earned. In the figure below, there are 8 petting stations (the nodes), numbered from 1 to 8. The links between stations shows that there are go-carts going between the two connecting stations and the number of zoo coins required to ride. The number next each node is the number of zoo coins earned after visiting that petting station. For example, after visiting station 1, 5 and 7, no additional zoo coins can be earned. However, visiting stations 2, 3, and 6 will earned 2 zoo coins. Riding go-cart to go from station 8 to station 2 and vise versa will cost 2 zoo coins. Given a starting petting station and initial number of zoo coins, determine the number of distinct petting stations that cannot possibly reached.

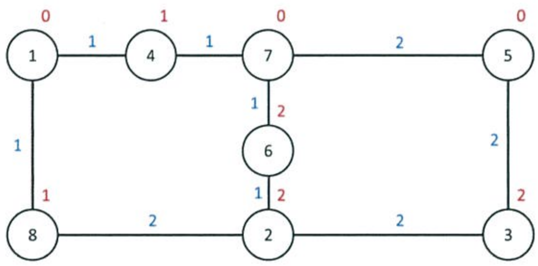

Figure 1: Petting Zoo.

For the figure above, a visitor arriving at station 1 with 1 zoo coin can reach petting stations 1, 4, 7, or 8, but not stations 2, 3, 5, or 6. However, if a visitor arrives at station 3 with 0 zoo coin, then any petting station is still reachable.

Technical Specification

1. There are at most 30 petting stations, numbered from 1, 2, ... 30.
2. Go-carts rides are available between some, possibly all, pair of petting stations. The cost of the ride is 1 or more zoo coins.
3. Each petting station may pay out 0 or more zoo coins after the visit.

Input File Format
First line of input is an integer indicating the number of test cases to follow. For each test case,

1. Line 1 consists of three numbers: the number of petting stations, the starting petting station and the number of zoo coins, all separated by a space.
2. Line 2 has a single integer p, indicating the number of pair of petting stations with go-cart services.
3. Line 3 includes p sets of 3-tuples $< S_i, S_j, c_1 >$, indicating that there is go-cart service between stations S_i and S_j with a cost of c_1 zoo coins.
4. Line 4 has a single integer z indicating the number of stations giving out non-zero zoo coins after the visit.
5. Line 5 includes z sets of 2-tuples $< S_k, c_2 >$, indicating that petting station S_k gives out c_2 zoo coins after the visit.

Output Format

For each test case, print the total number of stations that CANNOT be reached.

Sample Input

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

Output for the Sample Input

```
4
0
```

Problem C
School Hunting
Time Limit: *3 Seconds*

In the Great Country of Sky Dragon (GCSD), every junior high school student is required to take a high school education. There are high schools that many students want to attend. Since each school has a fixed quota for the number of students that can be enrolled, GCSD uses the following rules to decide which school a student goes to. Each junior high school student is required to take an exam with K subjects. The raw score of each subject is an integer that is at least 0 and at most M. The higher the score the better the score. The performance of a student depends on the score he/she obtained. The better the performance, the better the chance to get into his/her dream school. Let A and B be scores such that $M \geq A > B > 0$. Given the scores of a student α, let $A(\alpha)$ be the number of subjects whose raw scores are greater than or equals to A, let $B(\alpha)$ be the number of subjects whose raw scores are less than A and greater than or equals to B, and let $C(\alpha)$ be the number of subjects whose raw scores are less than B. Note that $A(\alpha) + B(\alpha) + C(\alpha) = K$.

The performance of student α is *better* than that of student β if and only if

- $A(\alpha) > A(\beta)$, or
- $A(\alpha) = A(\beta)$ and $B(\alpha) > B(\beta)$.

Note that in the case of $A(\alpha) = A(\beta)$ and $B(\alpha) = B(\beta)$, then $C(\alpha) = C(\beta)$ and they are equal in performance. A rank is then assigned using the above comparison scheme. The *rank* of a student is the number of students who perform better than him/her. Note that the rank of the best student is 0 and two or more students can possibly have the same rank. Each student has an unique ID that is an integer in the range of 1 to N. Given a list of N students' scores, we want to know the ranks of some S students.

Technical Specification

1. $0 < N \leq 10000$
2. $0 < K \leq 10$
3. $0 < M \leq 100$
4. $0 < A \leq M$
5. $0 < B < A$
6. $0 < S \leq 5$

Input File Format

The first line of the input file contains an integer, which is the number of test cases to follow. There are at most 10 test cases. For each test case, the first line contains 6 integers separated each by a space in the order of N, S, K, M, A and B. Lines 2 to $N+1$ each contains the scores of a student, so line $i+1$ contains the K subject scores, separated by space, of the

ith student. The next S lines, lines $N+2$ to $N+2+S-1$, each contains one student ID in which the rank is to be determined.

Output Format

For each test case, output the ranks of the S selected students in a line with a space in between two ranks.

Sample Input

```
2
4 4 3 7 5 1
4 7 7
5 5 5
1 5 5
0 7 7
4
3
2
1
3 2 5 100 80 60
100 90 76 25 78
79 79 79 79 100
99 99 100 23 45
2
3
```

Output for the Sample Input

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

Problem D
Sum and Product of Eigenvalues for a Square Matrix
Time Limit: *3 Seconds*

For an n by n real square matrix $A \in R^{n\times n}$, if there exists a non-zero $n-dimensional$ column vector $\mathbf{x} \in R^n$ such that $A\mathbf{x} = \lambda\mathbf{x}$. We call λ an eigenvalue of A and $\mathbf{x}$ is an eigenvector corresponding to the eigenvalue λ. It is obvious that any $A \in R^{n\times n}$ has exactly n eigenvalues which may not necessarily be all distinct.

The eigenvalues of a real square matrix $A \in R^{n\times n}$ plays an important role of applications in many areas like Anthropology, Biology, Engineering, Psychology, Zoology, and etc. It is well-known that not each eigenvalue of a real square matrix is real, whereas, the eigenvalues of every real symmetric matrix must be real. This problem asks you to design a program to compute the sum and product of a symmetric matrix AA^t for a given matrix $A \in R^{n\times n}$, where all of the entries of matrix A are integers in the range [-64, 64], and hence the sum and the product of the eigenvalues of AA^t are both integers.

We illustrate an example as follows.

$$A = \begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix}, \quad then \quad AA^t = \begin{bmatrix} 5 & 3 \\ 3 & 9 \end{bmatrix}, \quad and$$

$$\lambda_1 + \lambda_2 = 14, \quad and \quad \lambda_1 \times \lambda_2 = 36.$$

Input File Format

The first line of the input file contains one integer, $K \leq 5$, indicating the number of test cases (matrices) to come. The next K sets of $1+n$ lines consists of an integer n followed by n lines of n integer elements separated by space(s) in each line.

Output Format

There are K lines. Each line consists of the sum of eigenvalues and the product of eigenvalues separated by space(s).

Sample Input

```
 3
 2
 4  3
 0  4
 3
 3   0  0
 0   2  0
 0   0 -1
 5
-1  0  0  0  0
 0  2  0  0  0
 0  0 -3  0  0
 0  0  0  4  2
 0  0  0  2 -5
```

Sample Output

```
41  256
14  36
63  20736
```

Problem E
Acquisition Under Budget
Time Limit: *1 Second*

In the digital age, one of the most important mission of the National University Library is to subscribe digital academic resources for the students and faculty. There are a lot of valuable digital resources available in the market. Unfortunately, the Library has only a limited budget, and may not be able to subscribe all the resources. There are many strategies to determine which subset of these resources should be subscribed. In this problem, we use very simple strategy: the total subscription fee should be close to the budget, but not exceeding it.

Assume that the budget of the Library is m, and there are n academic resources with subscription fees $f_1, f_2, \ldots, f_n$. Write a program to find a subset of the academic resources whose total subscription fee is as close to m as possible, but not exceeding it.

Technical Specification

1. $m < 2^{20}$,
2. $n < 200$,
3. Each f_i is no more then m.

Input File Format

Each test case consists of 2 lines. The first line of the test case contains the integers m and n. The second line contains the n integers $f_1, f_2, \ldots, f_n$. The last test case is followed by a line containing 0.

Output Format

For each test case, print out the total fee to subscribe the academic resources.

Sample Input

```
100 5
62 36 79 51 36
5000 9
4339 1342 558 1033 4094 4220 417 275 4150
0
```

Output for the Sample Input

```
98
4983
```

Problem F
Noah2
Time Limit: *1 Second*

Humans, as predicted, totally destroyed the mother earth. Here comes Noah again to rescue various species. This time Noah's arch is a teleport machine called $arch2.0$. As you expected, some species do not get along well. They fight if Noah is not around. Noah carefully records those pairs, called them hostile pair. Even with teleport technology, the time to travel to and from mother earth to new planet is significant. Now Noah faces an agonizing problem, since he is the only one who can operate $arch2.0$, those species he left on earth or on the new planet must not hurt each other. Why not build $arch2.0$ to accommodate all the species all at once? You might wonder. The fact is that to build a larger $arch2.0$ takes a lot of time and efforts. According the species equality act, all species must be given equal space on any transportation. Therefore, the size of $arch2.0$ can be measured as the number of species that can be accommodated. For example, if there are 3 species A, B, C and A cannot get along with B nor can B get along with C. That is the set of hostile pairs are $\{(A, B), (B, C)\}$. The $arch2.0$ which can accommodate one species suffices to safely bring all species to the new planet. The way to do it is to bring B to the new planet, then you bring A to the new planet and bring B back to earth on the return trip. Then you bring C to the new planet. Finally you bring B to the new planet. Noah is busy leaping around the universe to save various species on different dying planets and desperately need your help. Given a set of species and their relationship to each other, Noah wants to know the smallest $arch2.0$ that can safely teleport all species. Each species is given an integer ID in the range of 1 to N. Given several endangered planets, you are to write a program to determine the smallest size $arch2.0$ for each case.

Technical Specification

1. $N \leq 10$.
2. There are at most 5 cases.

Input File Format

The first line of the input file contains an integer, n, which is the number of cases you have to solve. The following lines contains n cases. For each case, the first line contains 2 integers, v_i, e_i which are the number of species and the number of hostile pairs respectively. For the following e_i lines, each line contains 2 integers, which are the names of the hostile pair.

Output Format

For each test case, output the smallest size of $arch2.0$ that can complete the mission.

Sample Input

```
2
3 2
1 2
```

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

Output for the Sample Input

```
1
3
```