CSCI 2100B Data Structures

Midterm Examination (Programming Part)
6:00 p.m. - 9:00 p.m., Nov 8, 2013

Instructions

1. The programming midterm is an open-book and open-notes examination. You may bring what you can carry on printed (hard copy) materials. You MUST not take anything that can record program code electronically to the examination venue. You will not need a calculator for any calculation.

2. The operation system will be Ubuntu. The computer configuration will have these basic editors: vi/vim, emacs, gedit, and nano.

3. The examination will begin when the Chief TA starts the clock and will end when the Chief TA stops the clock, which is usually three hours after the starting time including any missing time due to technical and other difficulties.

4. You are suggested to work on Problem A first and then others afterwards. They are in increasing difficulties as judged by the instructor.

5. You MUST complete at least one problem in order not to fail the course.

6. Anyone who attempts to spam the server either through excessive submissions, allocating large amount of unnecessary memory, etc. will be penalized severely.

7. Please switch your mobile phones to silent mode, you are not allowed to use them during the exam.

8. If you want to go to the bathroom, please ask the TA for permission first.

9. If you leave early from the examination without informing the TA, you will not be able to come back to the examination.

Problem A - Password Security

When we set our passwords for a website, we care about the security of our account. Suppose we are only allowed with English letters and digital numbers. We say a password is "hard" if it contains both English letters and digits, and there are combinations of capital letters and lowercase letters. All other password are said to be "easy".

Input There may be several test cases in the input file. The first line is an integer N ($1 \leq N \leq 10$), which is the total number of passwords. In the following N lines, each line contains a string denoting a password. The length of each password will not exceed 256.

Output For each password, output 1 if it is a "hard" one; else, output 0.

Sample Input

```
4
Adgb123
abcd
1234
12abCd
```

Sample Output

```
1
0
0
1
```

Problem B - Integer Factorization

For years, people are interested about how to factor a composite number into primes. Here we try to do such a job with some small integers.

Input There may be several test cases in the input file. The first line is an integer N ($1 \leq N \leq 10$), which is the total number of integers to be factorized. In the following N lines, each line contains a positive integer, which is the number to be factorized. We guarantee that the input number is between 2 and 10^5.

Output Output all the prime factors of each input in the ascending order and in a single line. Each prime factor is separated with a blank space. There is no extra space in the end of each line.

Sample Input

```
4
12
17
100
150
```

Sample Output

```
2 2 3
17
2 2 5 5
2 3 5 5
```

Problem C - Longest Consecutive Sequence

Given an unsorted array of integers, find the length of the longest consecutive elements sequence. For example, given [100, 4, 200, 1, 3, 2], the longest consecutive elements sequence is [1, 2, 3, 4]. Thus the length will be 4.

Input There may be several test cases in the input file. The first line is an integer N ($1 \leq N \leq 100$), which is the total number of test cases. In the following N lines, each line contains an integer M ($1 \leq M \leq 10,000$), which is the number of elements in the array, followed by M positive integers in the range of $[1, 1000000]$. Each element is separated by a blank space.

Output For each test case, output the length of longest consecutive elements sequence in a single line.

Sample Input

```
3
4 1 1000 2 3000
6 100 4 200 1 3 2
10 87 234 34 84 10 11 95 85 12 235
```

Sample Output

```
2
4
3
```

Hints The range of elements is between 1 and $1,000,000$, which can be stored in the memory.

Source LeetCode Online Judge

Problem D - Multiplication

A great part of modern cryptography is built on big integer arithmetic. Here we are going to multiply two big integers which are much longer than 32-bits.

Input There may be several test cases in the input file. The first line is an integer N ($1 \leq N \leq 100$), which is the total number of test cases. In each case, there are two lines. The first line is a string denoting a positive big integer; the second line is the other positive big integer. Both integers are not larger than 10^{1000}.

Output For each test case, output the product of two integers as a string.

Sample Input

```
2
12345678978
10
123456789
123456789
```

Sample Output

```
123456789780
15241578750190521
```

Hints You may use a linked list or an array to store each big integer. Actually, a multiplication operation is just several addition operations.

Problem E - Couples

N couples are standing in a circle, numbered consecutively clockwise from 1 to $2N$. Husband and wife do not always stand together. We remove the couples who stand together until the circle is empty or we can't remove a couple any more. Can we remove all the couples out of the circle?

Input There may be several test cases in the input file. In each case, the first line is an integer N ($1 \leq N \leq 100,000$), the number of couples. In the following N lines, each line contains two integers, the numbers of each couple. $N = 0$ indicates the end of the input.

Output Output "Yes" if we can remove all the couples out of the circle. Otherwise, output "No".

Sample Iutput

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

Sample Output

```
Yes
No
```

Hints You may use a stack to simulate the removing process.

Source Sicily Online Judge

Problem F - Black Box

There is a very strange black box. In each day, the black box receives several elements. When the day ends, it removes the second largest element in the box and keeps all the remaining elements. It repeats these operations day after day. Now we want to know the sum of elements the black box removes after N days.

Input There may be several test cases in the input file. In each case, the first line is the total number of days, N ($1 \leq N \leq 50$), which is followed by N lines. In each line, the first integer M ($0 \leq M \leq 10,000$) denotes the number of elements the black box receives in that day, followed by the M elements. Each element is in the range of $[-32768, 32767]$. If on one day there is no element coming, the black box will still remove that second largest element. Assume that there are at least two elements at the end of each day so that you can always remove the second largest one. $N = 0$ indicates the end of the input.

Output Output the sum of elements removed in the N days in a single line.

Sample Iutput

```
5
3 1 2 3
2 1 1
4 10 5 5 1
0
1 2
0
```

Sample Output

```
16
```

Explanation At the end of the 1st day, the black box will remove 2. The remain elements are 1 and 3. On the second day, 1 and 1 will be added to the black box. It removes 1 at the end of the 2nd day. Following the process, the sum will be $2+1+5+5+3 = 16$.

Hints Although a max heap can only remove the largest element, you can use several operations to substitute the operation of removing the second largest element.

Problem G - Computer Network

A school bought the first computer (labeled 1) some time ago. During the recent years the school bought $N-1$ new computers (labeled from 2 to N). Each new computer was connected to one of settled computer earlier. Managers of school are anxious about slow functioning of the net and want to know for each computer number S_i - maximum distance, for which i-th computer needs to send signal (i.e., length of cable to the most distant computer). You need to provide this information.

Input There is natural number N ($2 \leq N \leq 10,000$) in the first line of input, followed by $N-1$ lines with descriptions of computers. i-th line contains two natural numbers - number of computer, to which i-th computer is connected and length of cable used for connection. Total length of cable does not exceed 10^9. Numbers in lines of input are separated by a space.

Output Write N lines in the output. i-th line must contain number S_i for i-th computer ($1 \leq i \leq N$).

Sample Input

```
3
1 1
1 2
```

Explanation The 2nd computer connects to the 1st computer with a cable of length 1. The 3rd computer connects to the 1st computer with a cable of length 2.

Sample Output

```
2
3
3
```

Source Saratov State University Online Contester