CSCI 2100B Data Structures

Midterm Examination (Programming Part)
9:30 a.m. - 12:30 p.m., April 13, 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 should 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 - String Revision

Given some lines of strings, you are asked to delete all the digits in the strings and print other characters in their original order.

Input There are multiple test cases. The first line of input is an integer T ($1 \leq T \leq 10$) indicating the number of test cases. Each case contains a line indicating a string. The length of the string is between 1 and 30, inclusive. Besides, the string consists of only digits and English letters.

Output For each string, delete all the digits and print the remaining characters in a line.

Sample Input

```
2
CSCI2100BDataStructure
1plus1equals2
```

Sample Output

```
CSCIBDataStructure
plusequals
```

Problem B - Intersection of Two Arrays

Given two array A and B, find out all elements both in Array A and B.

Input There are multiple test cases. The first line of input is an integer T ($1 \leq T \leq 10$) indicating the number of test cases. For each test case, first line is N ($1 \leq N \leq 1000$): the length of Array A. Next line contain N integers (in increasing order) in A. Next line is M ($1 \leq M \leq 1000$): the length of Array B. Next line contain M integers (in increasing order) in B. All elements k in A or B are guaranteed: ($1 \leq k \leq 10000$)

Output Numbers both in A and B at the same time, in increasing order, separate by one space. No space after the last number.

Sample Input

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

Sample Output

```
1 5
1
```

Problem C - Prime Palindromes

The number 151 is a prime palindrome because it is both a prime number and a palindrome (it is the same number when read forward as backward). Write a program that finds all prime palindromes in the range of two supplied numbers a and b ($5 \leq a < b \leq 100,000,000$); both a and b are considered to be within the range.

Input There are multiple test cases. The first line of input is an integer T ($1 \leq T \leq 10$) indicating the number of test cases. For each test case, there are two integers, a and b.

Output For each test case, output the list of palindromic primes in numerical order, one per line. There should be a blank line between test cases.

Sample Input

```
2
5 200
200 500
```

Sample Output

```
5
7
11
101
131
151
181
191

313
353
373
383
```

Problem D - Stone Pile

You have a number of stones with known weights $W_1, \ldots, W_n$. Write a program that will rearrange the stones into two piles such that weight difference between the piles is minimal.

Input There are multiple test cases. The first line of input is an integer T ($1 \leq T \leq 10$) indicating the number of test cases. For each test case, the input contains the number of stones N ($1 \leq N \leq 20$) and weights of the stones $W_1, \ldots, W_n$ (integers, $1 \leq W_i \leq 100000$) delimited by white spaces.

Output Your program should output a number representing the minimal possible weight difference between stone piles. One test case per line.

Sample Iutput

```
2
5
5 8 13 27 14
6
4 6 7 8 9 10
```

Sample Output

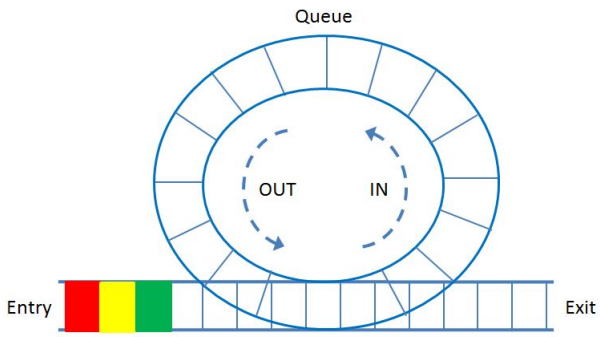
```
3
0
```

Problem E - Train Re-arrangement II

You are a railroad operator and you are asked to see whether you can re-arrange the carts in some order by using an auxiliary track, which can be regarded as a queue (See Figure below). The operations on the carts include the following:

- 'Straight Through', which means that you let the cart pass the main track directly without using the queue;
- 'Enqueue', which means that you put the cart into the queue;
- 'Dequeue', which means that you pull the top cart out from the queue.

The manage hopes you to save the operations as many as possible.



Input The input consists of the number of test cases, m in the first line and followed by m test cases. In each test case, the first integer is the total number of carts, n ($1 \leq n \leq 100,000$), followed by the order your manager wants you to achieve after your re-arrangement. Assume that in the initial state, all the carts are ordered from 1 to n, with Cart 1 in the first place.

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

Output The output should be m lines of operations. Each line should be the **shortest** sequence of 'S', 'I', and 'O' if you can achieve the demanded order by your re-arrangement. We use 'S' to denote 'Straight Through', 'I' to denote 'Enqueue', and 'O' to denote 'Dequeue'. There is no space between operations. If you cannot achieve the demanded order, please output 'Impossible'.

```
ISOS
Impossible
IIIISOOOO
```

Remark The problem is almost the same as what you have done in your homework **except** for the substitution from stack to queue.

Problem F - Fruit Merge II

The harvest season has come! Since Maggie works in an orchard, she has picked all the fruits and separated them into n piles according to the kind, e.g. one pile for apples, one pile for oranges, etc. After that, she wants to merge all these piles into one pile. Every time, she can only merge two piles, and the energy cost of merging them is the sum of weight of the two piles. It is easily to see that after $n-1$ merges, there is only one pile left, and the total energy cost is sum of energy cost in all these $n-1$ merges. Of course, Maggie wants to save her energy as much as possible.

Now, assume that the weight of each fruit is 1 and you have known the total number of piles and number of fruits in each pile. Your target is to output the minimum total energy cost of Maggie.

Hint It has been proved that you can achieve the minimum total energy cost by merging the two piles with the smallest number of fruits every time.

Input There are multiple test cases. In each test case, the first line is an integer n ($1 \leq n \leq 1,000,000$) representing the total number of piles. The next line includes n positive integers with each representing the number of fruits in each pile. Please note that we have sorted all these numbers in non-decreasing order for your convenience. The end of input is specified by a line in which $n = 0$.

```
3
1 2 9
6
1 1 3 4 4 6
0
```

Output For each test case, you should use one line to output your result. The result includes only one integer, which is the minimum total energy cost of Maggie after merging all the fruit piles into one pile. The answer will be fit in a signed 64-bit integer. You can use the type of **long long** to store it (use **%lld** in printf()).

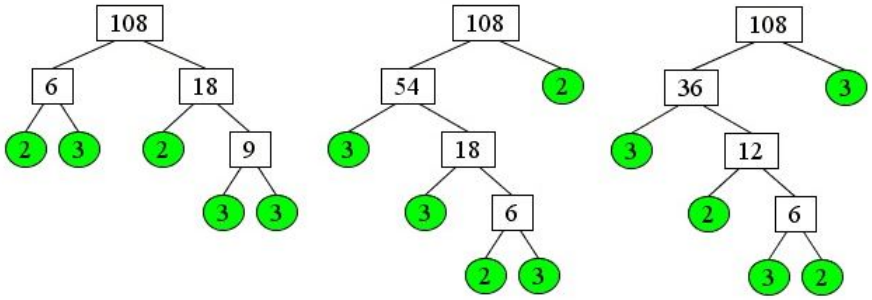
```
15
45
```

Remark The problem is exactly the same as what you have done in your homework **except** for the range of n. Please be careful on the time complexity of your algorithm.

Problem G - Counting Factor Trees

Factoring, i.e., listing all the prime factors, of an integer is a useful skill that often helps to solve math problems. For example, one of the ways to find the GCD (Greatest Common Divisor) or LCM (Least Common Multiple) of two integers is by listing all their prime factors. The GCD is then the product of all the common factors; the LCM is the product of all the remaining ones.

The Factor Tree is a tool for finding such prime factorizations. The figure below demonstrates three factor trees of 108. At the beginning a root with a number is given, say N, which is to be factored. Then, the root is factored into two children N_1 and N_2 such that $N = N_1 \times N_2 (N_1 \geq 2, N_2 \geq 2)$. Note that N_1 and N_2 need not be prime. The same factoring process continues until all the leaves are prime.



While the prime factorization is unique, the factor tree reflects the order in which the factors were found, and is by no means unique. So, how many factor trees of a number are there?

Input There are no more than 10,000 cases. A line containing an integer N ($2 \leq N \leq 1,000,000,000$) is given for each case.

```
12
108
642485760
```

Output Print the number of factor trees of N in a line for each case. The answer will be fit in a signed 64-bit integer. You can use the type of **long long** to store it (use **%lld** in printf()).

```
6
140
9637611984000
```