LEBANESE AMERICAN UNIVERSITY
School of Arts and Science
Department of Computer Science and Mathematics

CSC 310: Algorithms and Data Structures

Lab XI

Problem 1: [problem1.in]

Suppose you are given a set $S = \{a_1, a_2, ..., a_n\}$ of tasks, where task a_i requires p_i units of processing time to complete, once it has started. You have one computer on which to run these tasks, and the computer can run only one task at a time.

Let c_i be the completion time of task a_i, that is, the time at which the task completes processing. Your goal is to minimize the average completion time which is $(1/n) \sum_{i=1}^{n} c_i$. For example, suppose there are two tasks a_1 and a_2, with $p_1 = 3$ and $p_2 = 5$, and consider the schedule in which a_2 runs first. Then, $c_2 = 5$, $c_1 = 8$, and the average completion time is $(5 + 8) / 2 = 6.5$. If task a_1 runs first, then $c_1 = 3$, $c_2 = 8$, and the average completion time is $(3 + 8) / 2 = 5.5$.

Given a list of N items in which each item represents the completion time of a task, you are required to schedule the tasks so as to minimize the average completion time.

Your program will be tested on one or more test cases.

Each test case begins with an integer N, where $1 \leq N \leq 10000$
N integers follow, each representing the completion time of a task C, where $1 \leq C \leq 1000$.
The last line of the input file includes a dummy test case with $N = 0$.

For each test case, write the of the minimum average completion time up to two decimal places.

Sample Input	**Sample Output**
2 3 5	5.50
5 8 7 2 6 4	13.20
0	

Problem 2: [problem2.in]

Given a rod of length n inches and a table of prices p_i for $i = 1, 2, \ldots, n$, determine the maximum revenue r_n obtainable by cutting up the rod and selling the pieces. For example, assume we have a rod of length 4 and the following prices:

Length i	1	2	3	4
Price p_i	1	5	8	9

The optimal strategy here would be cutting the rod into two pieces of length 2, which yields a revenue of 10.

The first line of input is an integer T which is the number of test cases.

Each test case begins with an integer N, representing the length of the rod.
On a separate line, N integers follow, each representing the revenue for the length up to this integer.

For each test case, print the maximum revenue that can be achieved.

Sample Input	**Sample Output**
2 4 1 5 8 9 8 1 5 8 9 10 17 17 20	10 22

Problem 3: [problem3.in]

You have just begun working as a grocery bagger at the famous local "Khodarjov" food store. Your job is to place all of a customer's items into bags, so they can be carried from the store. Your manager has instructed you to use as few bags as possible, to minimize the store's overall cost. However, for the customer's convenience, you are instructed that only items of the same type can be placed in the same bag. For instance, a produce item can be bagged with any other produce items, but not with dairy items.

You are given a list of N items indicating the type of each item that needs to be bagged. You are also given S, the maximum number of items that can be placed in each bag. You should find the minimum number of bags required to package the customer's items.

Your program will be tested on one or more test cases.

Each test case begins with two numbers S, and N, where $1 \leq S \leq 50$, and $0 \leq N \leq 50$
N lines follow, each representing the type of item that needs to be bagged. An item type will be an upper case string with no spaces.
The last line of the input file includes a dummy test case with both $S = N = -1$.

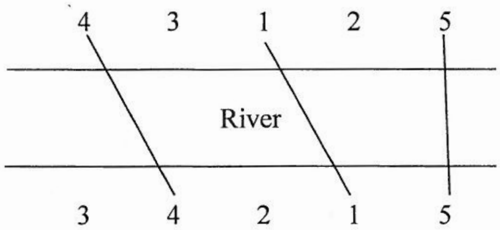
For each test case, write the result of the minimum number of bags required.

Sample Input	**Sample Output**
2 6 DAIRY DAIRY PRODUCE PRODUCE PRODUCE MEAT 3 6 DAIRY	4 3

DAIRY
PRODUCE
PRODUCE
PRODUCE
MEAT

Problem 4: [problem4.in]

Consider a 2-D map with a horizontal river passing through its center. There are *n* cities on the southern bank with x-coordinates a(1)... a(n) and *n* cities on the northern bank with x-coordinates b(1)... b(n). You are required to connect as many north-south pairs of cities as possible with bridges such that no two bridges cross. When connecting cities, you can only connect city *i* on the northern bank to city *i* on the southern bank. For example,



In the previous scenario, the maximum number of bridges we can achieve is 3.

The first line of input is an integer *T* which is the number of test cases.

Each test case begins with an integer *N*, representing the number of cities. The next line will have *N* integers, representing the cities on the northern bank, then another line of *N* integers representing the cities on the southern bank.

For each test case, print the maximum number of bridges that can be built.

Sample Input

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

Sample Output

3
2