

Bina Nusantara University
Bina Nusantara Programming Contest
for High-School Students 2006

Final Round
November 12th, 2006



PROBLEM SET

- A The Despot King
- B A Good Programmer
- C Cut The Line Chart
- D Danzel in Distress
- E Contest Rank List
- F Huffman Code
- G Rotation Pool
- H The Pirates of Greed Island

Instruction

You are given five hours to solve the problems in this contest. Input in each problem should be read from *input file*. Outputs on the other hand should be printed to *output file*. Both input and output file name should be the same as mentioned in each problem statement. You should submit your answer as an executable file (.exe) to the judge. Good Luck!

Problem A

The Despotic King

Input : king.in Time Limit : 5000ms
Output : king.out
Submit : king.exe

Problem Description

This is an old story of a country where Despotic King once ruled. There was a tradition for the king to give audience to all rich merchants in his country every year. On that important occasion, the minister was responsible to take each merchant into the audience room one after another and introduce the merchant's name and wealth. The merchant should wait in the room until all other merchants were gathered.

The king wanted the minister to announce the rank of each merchant in term of their wealth among all those who had arrived and waited in the audience room.

For example, suppose the set of wealth of 6 merchants who would meet the king in order of their arrival is:

78 24 68 40 39 89

The 1 st merchant would be ranked 1	<u>78</u> (there is no other merchant in the room)
The 2 nd merchant would be ranked 2	78 > <u>24</u>
The 3 rd merchant would be ranked 2	78 > <u>68</u> > 24
The 4 th merchant would be ranked 3	78 > 68 > <u>40</u> > 24
The 5 th merchant would be ranked 4	78 > 68 > 40 > <u>39</u> > 24
The 6 th merchant would be ranked 1	<u>89</u> > 78 > 68 > 40 > 39 > 24

Therefore, the minister would announce the sequence: 1 2 2 3 4 1

It would be troublesome since there were so many rich merchants in the country. Put yourself on the minister position and write a program to find the rank of each merchant.

Input Specification

The input file consists of several test cases. First line of each test case contains a single integer N ($1 \leq N \leq 10000$) indicating the number of merchants. The next N lines describe the wealth of each merchant in order of their arrival. Wealth should not be larger than 1,000,000,000 and there should be no negative wealth. All wealth should be distinct.

Output Specification

For each test case, print a single line contains "Case #x" (without quotes) where x is the case number. The next N lines follow, each describing the rank of each merchant when he/she enters the audience room (same order as the input).

Sample Input

6
78
24
68
40
39
89
3
24
59
36

Sample Output

Case #1
1
2
2
3
4
1
Case #2
1
1
2

Problem B

A Good Programmer

Input : good.in Time Limit : 1000ms
Output : good.out
Submit : good.exe

Problem Description

A good programmer must be good with numbers. That's why we would like to test your skill in numbers. Write a program to count how many pairs could be found where the difference between numbers in a pair is P .

For example, given the list of numbers below:

4 8 1 12 5

If the requested different is 4, then your program should give 3 as the output, where the pairs are 4-8, 8-12, and 1-5. Same numbers could be showed up more than once on the list. In this case, the numbers should be considered as different numbers.

Input Specification

The input file consists of several test cases. First line of each test case contains two integers: N ($1 \leq N \leq 10000$) indicating the number of items in the list, and P ($1 \leq P \leq 10000$) indicating the requested different. The second line contains N integer(s) $t_1, t_2, t_3, \dots, t_n$ ($1 \leq t_i \leq 10^8$) which shows the list of numbers.

Output Specification

For each test case, print in a single line a number of distinct pairs where the difference between numbers in pair is P .

Sample Input

```
3 2
3 1 6
5 4
4 8 1 12 5
4 3
4 1 1 4
```

Sample Output

```
1
3
4
```

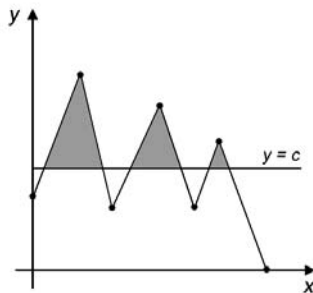
Problem C

Cut The Line Chart

Input : cut.in Time Limit : 1000ms
Output : cut.out
Submit : cut.exe

Problem Description

Calculating the area of a triangle might be an easy task for you. It can be done using a simple formula $\frac{1}{2}bh$, where b is the length of any side of the triangle (the base) and h (the altitude) is the perpendicular distance between the base and the vertex not on the base. Calculating any other shape should be easy with triangulation method.



Now, we would like to challenge you with a more difficult problem. Supposed you are given a series of line segments and a cut line $y=c$ in a Cartesian space, your task is to calculate all areas below the series of line segments and above the cut line.

The shaded area in the figure is describing the area to be calculated.

Input Specification

The first line of input is T the number of test cases. Each test case starts with two integers n ($2 \leq n \leq 100$) the number of points in the series, and c ($0 \leq c \leq 100$) the cut line equation that satisfies $y=c$. Then next n line each consists of two integers x and y ($0 \leq x, y \leq 100$) denoting point in Cartesian space. Two neighboring points form a line segment. All points will be given in the order of the line drawing. Each sequence of line segments will begin at $(0, y_1)$ where $y_1 \leq c$, and end at $(x_n, 0)$. The x -coordinate of the points will grow steadily.

Output Specification

You are to calculate how big an area below the line series and above the cut line is. The number should contain three digits after the decimal point.

Sample Input

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

1 0
2 2
3 0
4 2
5 0
9 1
0 0
0 4
1 4
1 3
2 4
3 3
3 2
4 3
4 0

Sample Output

1.000
1.250
9.500

Illustration for Case-1

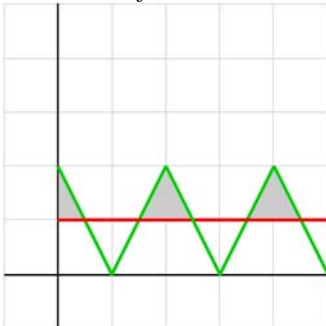


Illustration for Case-2

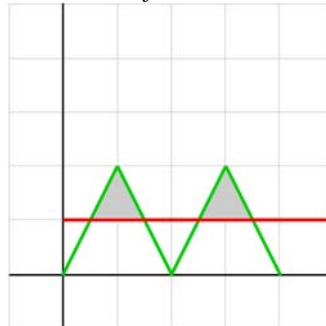
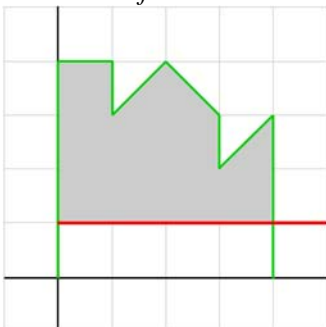


Illustration for Case-3



Problem D

Danzel in Distress

Input : danzel.in Time Limit : 1000ms
Output : danzel.out
Submit : danzel.exe

Problem Description

Previously on “A Simple Task”, you were helping Suhendry XIX to develop a program for a friend from a distant planet of Archeans, Archanos XIV of The Archean Intergalactic University (a.k.a AIU). Your program indeed satisfied him, he had successfully selected 3 top students from countless entities using your program without which he would have failed registering the students on time.

This time, Ms. Numanda Nexian, from The Faculty of Mathematics and Statistics of AIU has been asking for help. She has required you to develop a program to compute some statistical values from students' exam scores. The values to be computed are **maximum score**, **minimum score**, **average score**, **modus** (score that mostly occurs), and **median**. Because of the big number of students enrolling in AIU, the program must run efficiently.

For Clarity:

Assume a list of score { 5, 3, 4, 1, 2 }, the median is 3

For a list of score { 1, 5, 3, 4 }, the median is 3.5

Input Specification

Input file contains series of commands, each in one line. There are 2 types of command that should appear in the input file:

1. `ADD M` : add M to the list L . M is a student's score.
2. `VIEW STAT` : print the 5 statistical value of L in one line.

Constraints:

- L is assumed to be an empty list at the beginning
- M will be in the format `xx.xx`, ('x' should be between 0-9)
- The first line in the input file should never be `VIEW STAT`
- Input file contains no more than 10000 lines

Output Specification

For each `VIEW STAT` command, output 5 statistical value of the list L , **maximum score**, **minimum score**, **average score**, **modus** (score that mostly occurs), and **median** respectively. Each value should always be truncated to 2 digits behind the decimal point. For example, if the output is 49.999 then you should print 49.99, not 50.00. If there are more than one modus, output the largest one.

Sample Input

```
ADD 50.00
ADD 25.00
VIEW STAT
ADD 25.00
ADD 99.99
VIEW STAT
ADD 10.00
VIEW STAT
```

Sample Output

```
50.00 25.00 37.50 50.00 37.50
99.99 25.00 49.99 25.00 37.50
99.99 10.00 41.99 25.00 25.00
```


Problem E

Contest Rank List

Input : rank.in Time Limit : 1000ms
 Output : rank.out
 Submit : rank.exe

Problem Description

Mr. Rusli loves and has a good skill in algorithms. He has competed in BNPC-CS and won the sixth place for three years in a row (six must be his lucky number). Rather than wondering why he always got the sixth place, he interested in how many possible standing(s) (yes, we talk about the rank list) if there are N peoples in a contest.

Two simple rules apply:

1. There could be more than one contestants on the same rank.
2. For all $p > 1$, if rank p exists then rank $p-1$ have to be existed as well.

For example, if there are 3 contestants (A, B, and C):

	A	B	C		A	B	C
1 st Possible Rank :	1	1	1	8 th Possible Rank :	2	1	2
2 nd Possible Rank :	1	1	2	9 th Possible Rank :	2	1	3
3 rd Possible Rank :	1	2	1	10 th Possible Rank :	2	2	1
4 th Possible Rank :	1	2	2	11 th Possible Rank :	2	3	1
5 th Possible Rank :	1	2	3	12 th Possible Rank :	3	1	2
6 th Possible Rank :	1	3	2	13 th Possible Rank :	3	2	1
7 th Possible Rank :	2	1	1				

Given the example above, we can see that there supposed to be 13 possible standings for 3 contestants.

As an excellent *alorist*, Mr. Rusli would not need your help at all. Instead, he would like to test your skill in solving this problem. He wants you to write a program to count how many possible standing(s) would be if there are N contestants in a contest. Knowing that the result could be extremely large, he asks you only the last 4 digits of the result. Do not output any leading zeroes except the answer is zero itself. In that case, you should output only one zero instead of four zeroes.

Input Specification

Each input line contains a single integer N ($1 \leq N \leq 100$).

Output Specification

For each line of input, output the result (the last 4 digits) in one line.

Sample Input

1
2
3

Sample Output

1
3
13

Problem F

Huffman Code

Input : code.in Time Limit : 1000ms
 Output : code.out
 Submit : code.exe

Problem Description

Mr. Sanli has a problem, his computer's hard disk almost full and now he is going to decide whether he will buy a new hard disk or compress all his files. So he wants you to help him count how much spaces he can save by compressing his files.

To make your life easy, you only need to simulate this process in a string. The basic idea is replace binary representation for each letter (fixed-length code or ASCII code) in the word with *prefix-free codes*, which is shorter than the original one. Of course we would like to find the most efficient *prefix-free codes* mapping which able to represent the whole word.

The compressing steps are follows:

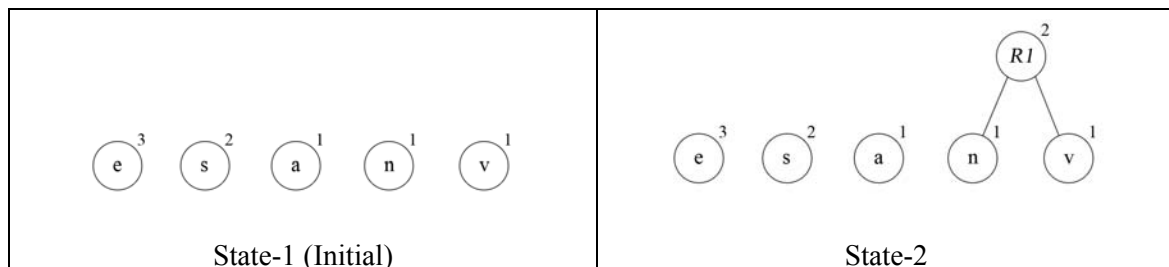
1. Count the occurrence frequency of each letter in the word. These letters along with their frequencies will be the leaves nodes/objects to construct following binary-tree.
2. Link two least-frequent objects with a new object -their parent node- whose frequency is the sum of those two frequencies. Now we will only consider this new object and ignore the other two.
3. Repeat Step-2 until we found the root objects.
4. In the tree we have constructed, the left node represent '0' and the right node represent '1'. For example, if letter e located on left-right-right from the root, than the prefix-free code for letter e is 011.

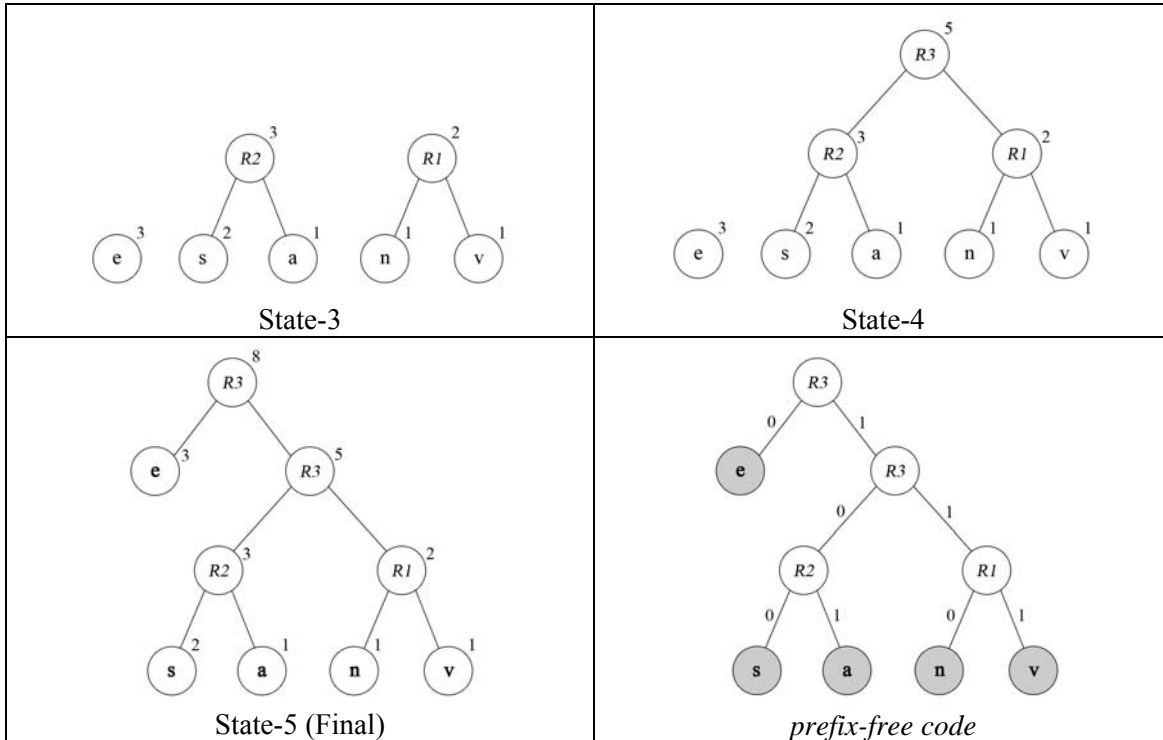
e.g.

Let the word `sevensea` to be compressed. First, we count the occurrence frequency of each letter.

`s = 2, e = 3, a = 1, n = 1, v = 1.`

The binary-tree constructions are described by the pictures below.





So, the *prefix-free code* for each letter would be:

e = 0 s = 100 a = 101 n = 110 v = 111

Normal string (ASCII code, letter in ASCII code represented in 8 bits)

01110011 01100101 01110110 01100101 01101110 01110011 01100101 01100001 64bit
s e v e n s e a

Compressed string (Huffman code)

100 0 111 0 110 100 0 101 18bit
s e v e n s e a

Input Specification

The first line of input is an integer T the number of test cases. Each test case contains one word (lowercase alphabet only) in a single line. Each word's length should not be more than 100 characters.

Output Specification

For each case, output the bit length of the compressed string using Huffman code method.

Sample Input

5
sevenssea
suryass
cuncun
stephen
suhendry

Sample Output

18
15
10
18
24

Problem G

Rotation Pool

Input : pool.in Time Limit : 1000ms
 Output : pool.out
 Submit : pool.exe

Problem Description

Rotation pool is one of the basic types of pool game (billiard) played by two players using 15 balls numbered from 1 to 15. The objective is to collect as many points as you can by pocketing balls. If you pockets ball number p then you will get p points added to your total score. A player gets his turn if the other player made a foul or failed to pocket any ball in his shoot. The game ends when one of the player's score is more than 60 which is more than half of the available points, and so he is decided as the winner.

In the example below, Player A manages to pocket balls number 1, 2, 4, 7, 11, 12, and 13. In the other hand, Player B manages to pocket balls number 3, 5, 6, 8, 9, 10, 14, and 15.

Player	Pocketed Balls														Total	
A	1	2		4			7				11	12	13			50
B			3		5	6		8	9	10				14	15	70

So, the total score for A is $1 + 2 + 4 + 7 + 11 + 12 + 13 = 50$, and for B is $3 + 5 + 6 + 8 + 9 + 10 + 14 + 15 = 70$. That means this game is won by Player B who leads by 20 points.

Given the data of which ball pocketed by each player, write a program to calculate the difference between the winner score and the loser score. In this problem, we will assume the game still continued even if there is a player who scores more than 60.

Input Specification

Input consists of several test cases. Each test case contains two lines. The first line contains an integer N ($0 \leq N \leq 15$), the number of balls pocketed by first player. The second line contains 15 integers (1 to 15) each separated by a single space. The first N integers are the balls pocketed by first player, and the rest $15-N$ integers are balls which pocketed by the second player.

Output Specification

For each test case, output the difference between the winner score and the loser score in a single line.

Sample Input

```
7
1 2 4 7 11 12 13 3 5 6 8 9 10 14 15
6
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
```

Sample Output

```
20
30
```

Problem H

The Pirates of Greed Island

Input : pirates.in Time Limit : 1000ms
Output : pirates.out
Submit : pirates.exe

Problem Description

Once upon a time, there was a band of well known pirates of the seven seas, The Pirates of Greed Island. Each member had a unique rank which decided his authority over others. The highest rank was the leader himself.

Though they were pirates, they applied a democratic system to distribute their gold coins. The leader decided how the coin should be distributed to each member. If there were more than half of the pirates (include himself) rejected his decision, than he would be punished and thrown into the deep sea, which meant he was killed. And the next new leader would be the highest rank among all living pirates. This new leader then should make the next decision. This process continued until the decision was accepted by at least half of the living pirates (including the leader).

A pirate could reject a decision if he could earn more coins by rejecting the decision. Of course any leader did not want to die, but he also wanted to gain as many coins as he could. Sometimes, the leader could not escape from his death despite of what he decides.

It seemed computer had not been invented yet at that time, but we did not care, you are about to simulate this system and calculate how many maximum coins can be earned by the leader of all living pirates.

Input Specification

Input consists of several test cases. Each case contains two integers N ($1 \leq N \leq 10000$) indicating the number of pirates and P ($1 \leq P \leq 10000$) indicating the number of gold coins.

Output Specification

For each test case, output in a single line an integer E indicating the maximum amount of coins could be earned by the living pirates' leader.

Sample Input

```
2 100
3 10
```

Sample Output

```
100
9
```