ECOO 2017

Programming Contest Questions

Local Competition (Round 1)

March 29 – April 5, 2017

Problem 1: Munch ’n’ Brunch

Author: Andrew Seidel

Student council is looking to organize a school brunch, where the proceeds will be put towards a year-end trip for the graduating class. The council members decide that the price depends on how many years you have been at the school. For someone who has been at the school for one year (**Y_1**), the price will be \$12, for someone who has been at the school for two years (**Y_2**), the price will be \$10, the three-year (**Y_3**) price will be \$7, and the price for someone who has been there all four years (**Y_4**) will be \$5.

Out of all the proceeds, 50% can be saved towards the year-end trip, as the other 50% is spent on the various costs to run the brunch. Given the following input data, calculate whether or not the council will need to raise additional funds.

Input Specifications

DATA11.txt (DATA12.txt for the second try) will contain 10 trips, at 3 lines of data per trip.

- For each of the trips, the first line will show the cost of the trip as an integer (\$50 to \$50,000).
- The next line contains four floating point numbers **Y_1**, **Y_2**, **Y_3**, **Y_4** ($0 \le Y_1, Y_2, Y_3, Y_4 \le 1$ and $Y_1 + Y_2 + Y_3 + Y_4 = 1$) representing the percentages of the total number of students from years 1 through 4 respectively.
- The third line contains a single number, **N**, which contains the total number of students attending the brunch ($4 \le N \le 2000$)

Note: You cannot have less than a whole person (e.g., 1.8 people is the same as 1 person). Any missing or extra people should be removed from or added to the group with the highest percentage of attendees. There will always be exactly one group with the highest percentage of attendees.

Output Specifications

Output “YES” if the student council needs to find other funding, and “NO” if the council has raised sufficient funds.

Sample Input

```
4000
0.5 0.2 0.1 0.2
400
6000
0.1 0.1 0.45 0.35
2000
```

Sample Output

```
YES
NO
```

Problem 2: Chocolate Chewsday

Author: Andrew Seidel

At the local candy factory, every Tuesday, there is a contest for whoever can come up with the best new chocolate. To make a decision on whether or not each chocolate is a winner, there is a panel of impartial judges that come in from the community. The judges are given the following criteria for judging:

- Packaging (**P**), up to 1 point
- Flavour (**F**), up to 2 points
- Minimal ingredients (**G**), up to 3 points

A score, **S**, is given to each chocolate from each judge assigned to that chocolate (0 ≤ **S** ≤ 6). There will be a random number of judges, **J**, assigned to each chocolate (1 ≤ **J** ≤ 100).

Your task is to declare the winner based on the highest total score in the competition. If there is a tie for highest total score, it can sometimes be broken using the total scores for **P**, **F** and **G** (try **G** first, then **F**, then **P**).

The competition is not really fair because some chocolates get more judges than others. But that's life at the candy factory.

Input Specifications

DATA21.txt (DATA22.txt for the second try) will contain 10 competitions. The first line of each competition will contain a single integer, **N**, to indicate the number of chocolates in the competition (1 ≤ **N** ≤ 100). For each of the **N** chocolates, there will be **J**+1 lines in the file. The 1st line is the name of the chocolate (a single word with no spaces) and the next **J** lines will contain the judges' scores (1 ≤ **J** ≤ 100). Each score will be contained on a single line, starting with the letter **J** followed by the 3 integers **P**, **F**, and **G** separated by spaces. Each competition ends with an asterisk (*).

Output Specifications

Output the name of the winner. If there is more than one winner, print out all winners on a single line separated by commas (order does not matter – i.e., an output of A, B is the same as B, A).

Sample Input (only 1 case shown)

```
2
C1
J 0 1 1
J 0 1 0
J 1 0 0
C2
J 1 2 3
*
```

Sample Output

```
C2
```

(See next page for another example)

Sample Input (only 2 cases shown)

```
4
ChocolateOfChocolates
J 0 2 2
J 0 1 2
J 1 2 0
Choco-Fun
J 1 2 3
J 1 2 0
ChocolateHaven
J 1 2 0
J 0 2 3
J 1 0 1
ChocolatesRock
J 1 2 1
J 1 2 0
J 1 2 0
*
1
ChocolateFilledCandy
J 0 0 0
*
```

Sample Output

```
ChocolatesOfChocolates
ChocolateFilledCandy
```

Explanation

For the first competition, there is a tie between *Chocolate of Chocolates*, *Chocolate Haven*, and *Chocolates Rock*. We had to then look at the **G** values, which were tied for *Chocolate of Chocolates* and *Chocolate Haven*. Consequently, we had to then check the **F** value. At this point, *Chocolate of Chocolates* has the higher value.

ECOO 2017 Question Development Team

Kevin Forest	Sheridan College
John Ketelaars	ECOO-CS Communications
Stella Lau	University of Cambridge
Greg Reid	St. Francis Xavier Secondary School, Mississauga
Sam Scott	Mohawk College
Andrew Seidel	John Fraser Secondary School, Mississauga
David Stermole	ECOO-CS President
Reyno Tilikaynen	University of Waterloo

Problem 3: Mountain View

Author: Reyno Tilikaynen

On his trip to the West Coast, Larry has decided to stop by the Rocky Mountains to do some sightseeing. He only wants to climb one mountain, so he would prefer to climb the mountain with the best view.

The quality of the view is judged by the number of other mountains Larry can see from the peak when looking either right or left. A mountain is visible from a peak if the straight line connecting the peaks of the two mountains doesn't touch any other mountains.

Given the heights of all the mountains in the range, can you figure out which mountain has the best view? The mountains are all lined up in a straight line and spaced 1 unit apart.

Input Specifications

DATA31.txt (DATA32.txt for the second try) will contain 10 test cases. Each test case starts with an integer **N** (1 ≤ **N** ≤ 10,000) indicating the number of mountains. The next line contains **N** integers ranging from 1 to 10,000, representing the heights of the mountains in the range. Mountains are numbered from 1 to **N**.

Output Specifications

For each test case, your program should output an integer corresponding to the mountain with the best view. If there is a tie, output the smallest numbered mountain.

Sample Input (only 3 cases shown):

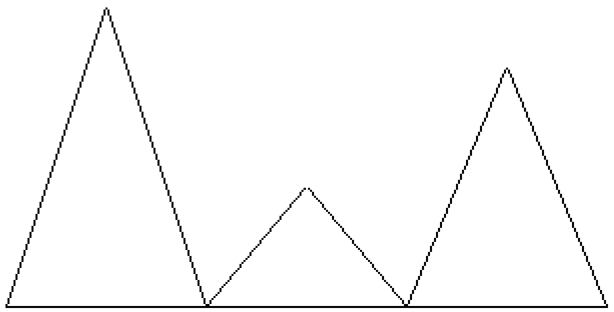
```
3
5 2 4
5
5 2 4 3 3
5
5 4 1 1 1
```

Sample Output:

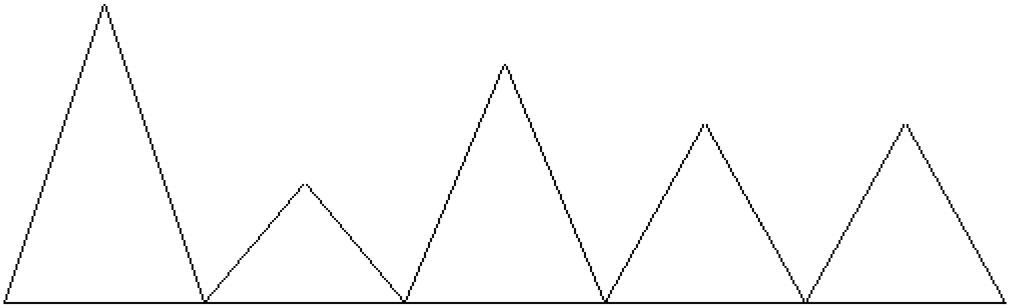
```
1
3
2
```

Explanation

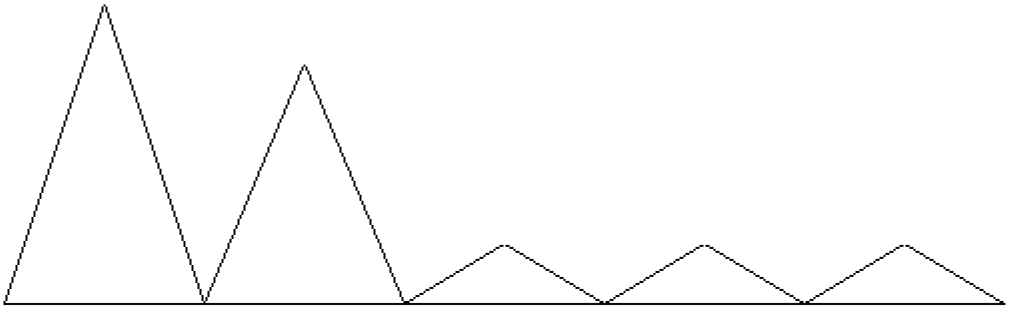
In the first case, all three mountains have views of the entire range.



In the second case, the first mountain is the highest. However, you can’t see the last two mountains from it since the third mountain is in the way. The third mountain has a view of the whole range.



In the third case, the view from the first mountain only contains the second mountain. The second mountain has a view of the whole range.



Problem 4: Almost Sorted

Author: Reyno Tilikaynen

After the lecture on sorting, Diana's CS teacher has given the class a challenge: what's the most efficient way to sort the class list? Students are only allowed to sort by repeatedly swapping two names in the list, and the person who makes the fewest swaps wins.

Diana really wants to win, so she decides to do something sneaky. When showing the teacher her efficient sort, she will omit one of the names in the list, which will let her use fewer swaps. Using her tactical advantage, what is the fewest swaps she has to do?

Input Specifications

DATA41.txt (DATA42.txt for the second try) will contain 10 test cases. Each test case starts with an integer **N** (1 ≤ **N** ≤ 1,000). The next **N** lines represent the class list, with one name on each line in capital letters. Names will be unique and only contain uppercase letters.

Output Specifications

For each test case, your program should output the minimum number of swaps Diana needs to sort the list using her trick.

Sample Input (only 2 cases shown)

```
3
SAM
DIANA
REBECCA
5
DEREK
MEGAN
BRIAN
BOB
DIANA
```

Sample Output

```
0
1
```

Explanation

In the first example, removing "Sam" will leave the list sorted, so no swaps are required. In the second example, Diana could omit "Megan" and then sort the list by swapping "Derek" and "Bob".