SWE2026 Problem Solving Techniques, 2021 Spring

Instructor: Younghoon Kim

PA 2: FC PST 2, Mask Please

TA in charge of FC PST 2: Soohwan Kim, Mingyu Choi

TA in charge of Mask Please: Mingyu Choi, Sungmin Kang

1. Submission Details

1-A. Scoring Criteria

- 90% for Programming, 10% for Report.
- Programming is directly scored by *goorm* platform.
- Report is P/F based.
    - ■ If your report has less than 4 issues (over-volume, under-explanatory, etc.), grade **P** (Pass, full credit). Otherwise, **I** (Incomplete).
    - ■ After you get **I**, you can submit once more. If it is okay, grade **1/2 P** (half Pass, half credit). Otherwise, **F** (Fail, no credit).

1-B. Deadline

- **Ten days** for regular submission, two days for delayed submission.
- Regular submission due for both program code (via *goorm*) and report (via *iCampus*) is **2021/04/07(Wednesday) 23:59:00.**
- Delayed submission penalty is 25% per day, and submission will be closed on 2021/04/09(Friday) 23:59:00.
- Be careful: *goorm* **do NOT warn** for delayed submission and penalty is automatically applied, and the last one minute of the final day is considered as late.

1-C. About Report

- Either **Korean or English** is acceptable.
- **Should keep the line limits**. Make your answers clearer by shortening them.
- **Do NOT handwrite** your answers. Only figures can be hand-drawn in empty spaces.
- **HANDWRITE** the signature of honor code. Writing via pen (on tablet PC) is okay.

2. Problem 1 - FC PST 2

2-A. Problem Description

** This problem is based on understanding of previous assignment, PA 1-1. FC PST.

Mr. Kang, the manager of FC PST, has been keeping all the game records of his team. The way he wrote match results and calculated ASGD was as follows:

- He writes the date(month and day) of that day, GF(Goal For, or earned score) and GA(Goal Against, or lost score).
- The pages that he writes match results are randomly selected.
- He calculates ASGD(accumulated sum of goal difference) on a daily basis, indicating that he only cares about the total results of each day when his team has to play several games a day.
- Goal difference(GD) is the difference between scoring and losing(GF - GA). Of course, GD can be a negative value.

As he has been using the method above in managing his team, he found out that he could get more precise ASGD if he changes his way of writing and calculating scores.

Here are some changes that he tried to make:

- To cover a wider range of period, add year at the beginning of date. (i.e. 2021 3 27 3 1, meaning FC PST beat 3-1 in a game of Mar 27th, 2021)
- To get an immediate and precise result, calculate ASGD on a match basis indicating that he calculates it for every single game.

One problem, however, is that Kang cannot remember the order of games in each day because the match notes does not specify the exact time of the game. In that sense, he decided NOT to care about the order of the games when calculating ASGD for one day.

Considering a few changes above, find out the earliest date on which the |ASGD|(the absolute value of ASGD) reached its highest point and its value at that moment in this situation.

2-B. Restriction

In the first line, the number of notes(or match results) that Kang wrote, N is given.

For next N lines, the details(Year_i, Month_i, Day_i, GF_i, GA_i) of each note are given separated by one space.

Print the earliest date(Year, Month and Day) when the absolute value of ASGD (Accumulated Sum of Goal Difference) is the largest and the |ASGD| at that time, separated by one space.

Details

- For all test cases, Year_i, Month_i, Day_i are valid value on common year (you do not need to consider leap year), which means all years have 365 days, so no matches can take place on 29 February.

- For all test cases, 1,900 <= Year_i <= 2,500 and N <= 500,000
- For 50% of test cases, Year_i = 2,020 and N <= 1,000 and (number of games played in a day) <= 3
- For 20% of test cases, N <= 10,000 and (number of games played in a day) <= 7
- For the rest 30% of test cases, no limitations on input range.

2-C. Example

Input	Output

4
2020 2 11 1 2
2021 6 29 5 1
2021 6 29 0 3
2021 8 21 0 2
	2021 6 29 4

Description

- 2020-02-10: the ASGD is 0 because not a match took place at all.
- 2020-02-11 ~ 2021-06-29: the ASGD is -1.
- Now we have two possible scenarios:
  1) take ‘2021 6 29 5 1’ as preceded event
  2) take ‘2021 6 29 0 3’ as preceded event
  We can find out that 2) would be the best choice because it leads to larger |ASGD| in the end. (|+3| < |-4|)
- 2021-06-29: After game of 2), the ASGD is -4. After game of 1), the ASGD is 0.
- 2021-06-30 ~ 2021-08-20: the ASGD is 0.
- 2021-08-21 and after: the ASGD is -2.

The answer is 29th June, 2021 and ASGD of -4, thus you should print ‘2021 6 29 4’.

3. Problem 2 – Mask Please

3-A. Problem Description

Due to a highly contagious virus, the Korean government announced new regulations on all indoor business operators to check if their customers wear masks.

A diligent citizen, Frank is running an indoor study cafe in Korea. He is trying to check whether each individual customer is wearing a mask. However, he is too busy managing other businesses to keep his eye on visitors all day long. In this situation, he wants to know the minimum number of times that he needs to come to check his cafe.

Suppose we can know when each customer will arrive and leave the cafe in advance. Help him make a program which calculates the minimum number of times to check each customer.

3-B. Restriction

In the first line, the number of people visiting the café is given.

For the next N lines, the times for entering and exiting, Time_in and Time_out respectively, are given in pairs for each person.

Print the minimum number of times needed to check all customers at least once.

Details

- For all test cases, non-negative integers Time_in and Time_out are equal or less than 1,000,000,000
- 0 <= N <= 500 000
- For 50% of test cases, Time_in and Time_out <= 1,000 and N <= 1,000
- For 20% of test cases, no limitations on time and N <= 10,000
- For the rest 30% of test cases, no limitations on input range.

3-C. Example(s)

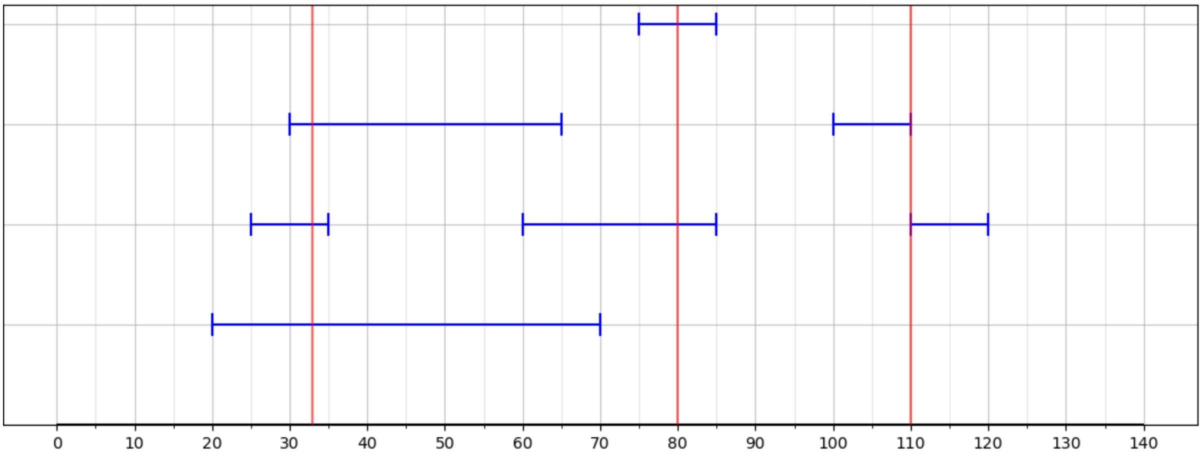
Input	Output

3
28 67
54 90
71 82
	2

7
60 85
20 70
100 110
25 35
75 85
30 65
110 120
	3

Description

- On the first line, we get the number of the people visiting the café (here 3)
- The following lines specify the entering time and exiting time of each individual person
- Person one enters at 28 and exits at 67
- Person two enters at 54 and exits at 90
- Person three enters at 71 and exits at 82
- Person one and three are never in the café at the same time (their times do not overlap)
- Person two overlaps both with Person one and two
- The owner can either check Person one and two or Person two and three at the same time as a maximum
- Hence, one additional check is necessary to verify that every customer has worn a mask.
- Therefore, the minimum amount of the owner having to check is 2.
- For the second example, please refer to the picture below.
- Blue lines are for people, and red lines are for the owner checking.


0
10
20
30
40
50
60
70
80
90
100
110
120
130
140