PA 3 (Mid-term Exam): Scholarship Officer

TA in charge of: Sungmin Kang, Mingyu Choi, Soohwan Kim

1. Submission Details

1-A. Scoring Criteria

- **50%** for programming results in *goorm*, **20%** for specific testcases scored by TA and the rest of **30%** for report.

- 50% of the score for Programming is directly scored by *goorm* platform.

- 20% of the score is graded by TA with our own testcases after the late submission due.

- You do not need to submit your codes for the 20% portion. We will directly use your codes that have been submitted to *goorm*.

- **Note that the report of this PA will be graded in detail. Please pay more attention to given instructions this time.**

  - The report for PA3 includes new forms of problems.

  - In PA3, your score of the report will be ranked.

  - Please check appendixes in the report. They provide additional examples and details.

1-B. Deadline

- **Ten days** for regular submission, and two days for delayed submission.

- The regular submission due for program code (via *goorm*) and report (via *iCampus*) are both **2021/04/28(Wednesday) 23:59:00.**

- Delayed submission penalty is 25% per day, and submission will be closed on 2021/04/30(Friday) 23:59:00.

- Be careful: G*oorm* **do <u>NOT</u> warn** for delayed submission and penalty is automatically applied, and the last one minute of the final day is considered as late.

- **From PA3, we will not give you any excuse for late submissions by mistake.**

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 fine.

2. Problem Definition: Scholarship Officer

2-A. PA Description

- This PA consists of **two subproblems**.

- The *Goorm* programming scores for both problems are equivalent.

- Both the problem scores of TA specific testcases will be evaluated by running time of codes and ranked by their average.

- The problems are identical in a way that they **share one problem description**.

- The only difference between these subproblems is **data distribution**.

- You may use different or same codes for each subproblem.

- Report is not separated between subproblems, meaning that you should **submit one report only**.

- If you use same codes for each subproblem, you have to mention relevant information in the report.

2-B. Problem Description

SungKyunKwan Univ. (SKKU) and PST foundation are establishing DB systems for managing scholarship recipients. Although both of their purposes are the same (finding students who are eligible for scholarships), those two systems are under different circumstances.

Firstly, the **SKKU** system has a lot more students compared to the number of DB queries. At the time of admission, a large number of students are newly registered to be candidates. However, SKKU not only has a few scholarships due to its limited budget but also provides scholarships to relatively broader range of students than PST foundation does. It results in a smaller number of queries considering the size of DB.

In case of scholarships provided by the **PST Foundation**, on the other hand, the number of students registered in DB is comparatively small because students are filtered through the registration process. The foundation considers various aspects of students such as GPA, volunteer activity, etc. Despite its small size, DB of the foundation still has more queries to be processed to find suitable candidates because the scholarship beneficiaries are chosen even more carefully. Therefore, its DB generally handles more DB queries than SKKU DB does.

Please help SKKU and PST foundation to implement efficient DB systems under the consideration of their own characteristics. The followings will be used for categories of the data and query.

- **Major:** SKKU has Natural Science, Engineering, Software, Humanities and Social studies, Arts and Academic colleges, and all students must belong to one of these **6** departments. (Each department is expressed as a value of 1,2, ...,6.)

- **Semester:** Students are classified according to the number of semesters they attended. Attended semesters of each student range from **1** to **10**.

- **Credits:** Students can take **1** to **18** credits per semester.

- **Number of satisfied graduation requirements:** Students who meet **3** out of 5 graduation requirements are eligible for graduation.

- **Total scholarship amount:** The total scholarship amount that each student can receive is less than or equal to **200 000 000**. All inputs are given **sorted in an ascending order** based on total scholarship benefits.

- **Query types:** Query has three types that are described as follows.

  1. *FIND* : Find if students with a specific Total Scholarship Amount (TSA) exists.

  2. *HIGHER* : Find the number of students with a specific TSA or higher.

  3. *LOWER* : Find the number students with a specific TSA or lower.

- **Categories in Queries:** The queries can specify the categories of the students. Therefore, each query states the condition of target students for each category. Target of single category is two types, described in the following.

  1. No condition: Target all students regardless of the value of this category.

  2. One specified: Target only students that have one specified value of this category.

Implement a DB system that stores students' data according to the conditions above and handles queries to find the number of students who fall under specific conditions.

3. Subproblem Definition: Scholarship Officer

3-A. Subproblem 1 – Many Students

In this subproblem, the number of students is larger than the number of queries.

N refers to the number of students, and M to the number of queries.

Details

- For all test cases, 1 <= M < N, 0 <= TSA <= 200 000 000

- For 30% of test cases, there are no 0 in queries, N <= 1 000, M <= N / 2 and TSA < 100 000

- For other 20% of test cases, N <= 10 000, M <= N / 4, TSA < 10 000 000

- For rest 50% of test cases, N <= 1 000 000, M <= N / 64

3-B. Subproblem 2 – Many Queries

In this subproblem, the number of queries is larger than the number of students.

N refers to the number of students, and M to the number of queries.

Details

- For all test cases, 1 <= N < M, 0 <= TSA <= 200 000 000

- For 30% of test cases, there are no 0 in queries, M <= 1 000, N <= M / 2 and TSA < 100 000

- For other 20% of test cases, M <= 10 000, N <= M / 4, TSA < 10 000 000

- For rest 50% of test cases, M <= 1 000 000, N <= M / 32

3-C. TA Specific Test Cases

In TA specific test cases, there are also 2 subproblems that are the same as above.

There are 5 input cases for each subproblem.

N is for the number of students, and M is for the number of queries.

Details

- For Many Students, N <= 100 000 000, M <= N / 128 and there are some edge cases.

- For Many Queries, M <= 100 000 000, N <= M / 128 and there are some edge cases.

- If your program gets into following situations for any reason, your running time will be considered as 1500 seconds for each case.

    - The running time exceeds 5 minutes.

    - Memory utilization exceeds 1GB.

    - Compilation or Runtime Error

    - Print wrong answers.

4. Restriction: Scholarship Officer

4-A. Input & Output

In the first line, the number of students(N) is given.

For the next N lines, each student's category values and TSA are given.

They are given in order of Department, Semester, Credits, Satisfied Graduation Requirements, TSA.

And then, the number of queries(M) is given.

For the next M lines, each query's type, category values and TSA are given.

They are given in order of Query Type, Department, Semester, Credits, Satisfied Graduation Requirements, TSA.

Print results of each query in a given order, separated by line breaks.

- For query type 1(FIND), print "1" if a student who satisfies all conditions exists and "0" if not exists.

- For query type 2(HIGHER) and 3(LOWER), print the number of students found.

4-B. Data Range

- N and M follow the restrictions as described in its own subproblem details.

- For student categories(or SC),
  $1 <=$ first category (SC_1) $<= 6$,
  $1 <=$ second category (SC_2) $<= 10$,
  $1 <=$ third category (SC_3) $<= 18$,
  $0 <=$ fourth category (SC_4) $<= 5$

- $0 <=$ TSA $<= 200\ 000\ 000$

- For all SC_*s and TSAs are in above ranges. TSA of students are in ascending order.

- $1 <=$ Query Type (QT) $<= 3$

- For QT, value 1, 2, 3 means query of 'FIND', 'HIGHER', and 'LOWER' respectively.

- For query categories(or QC),
  $0 <=$ first category (QC_1) $<= 6$,
  $0 <=$ second category (QC_2) $<= 10$,
  $0 <=$ third category (QC_3) $<= 18$,
  $0 <=$ fourth category (QC_4) $<= 2$

- 0 in QC_* means 'No condition' of that category, meaning you should target all students of this category.

- For QC_1, QC_2, QC_3, the value of SC_1, SC_2, SC_3 respectively is 'One specified' of that category. It means that target students who have exactly same value of this category.

- For QC_4, value '1' means 'One specified' which specifies students who are eligible for graduation (SC_4 >= 3), whereas value '2' means 'One specified' which specifies students who are not eligible for graduation (SC_4 < 3).

- All QTs and TSAs are in above ranges. TSA of queries are unordered.

- You do not need to care about wrong inputs. (i.e. Input values out of ranges above.)

5. Example: Scholarship Officer

5-A. Example 1

Input	Output
5 1 1 3 2 100000 5 1 18 4 3000000 1 6 12 2 25000000 1 6 12 5 28000000 4 6 16 1 32000000 3 2 0 0 0 0 25000000 1 0 6 12 1 28000000 3 0 1 0 2 200000	3 1 1

Description

- There are 3 students whose TSA is higher than or equal to 25 000 000.
  - 3rd, 4th and 5th students.
- There is a student who takes 12 credits in the 6th semester and satisfied graduation requirements and whose TSA is exactly 28 000 000.
  - 4th student.
- There is only 1 student who is in the first semester and did not satisfy graduation requirements and whose TSA is lower than or equal to 200 000.
  - 1st student.

5-A. Example 2

Input	Output
5 5 1 6 2 500000 5 1 6 2 1500000 1 1 6 3 20000000 1 1 6 3 58000000 1 1 6 3 62000000 3 2 0 0 0 0 10000000 1 0 1 6 0 200000 3 1 0 0 1 35000000	3 0 1

Description

- There are 3 students whose TSA is higher than or equal to 10 000 000.
  - 3rd, 4th and 5th students.
- There is no student in the database who takes 6 credits in the 1st semester and whose TSA is 200 000.
- There is a student who belongs to the Natural Science and satisfied graduation requirements and whose TSA is lower than or equal to 35 000 000.
  - 3rd student.