

Input:

before peer review

student [StudentID] [StudentID] ...

schoolStrategy [Level],[Score] [Level],[Score] ...

peer review

designCriterion [AssignmentID] [RubricFile]

assignment [AssignmentID] [StudentID] [ReviewrID],[SampleScoreFile] [ReviewrID],[SampleScoreFile]...

after peer review

printRubric [AssignmentID]

averageCriterion [AssignmentID]

calculateScore [AssignmentID] [StudentID] [RankingStrategy]

findStrength [AssignmentID] [StudentID] [RankingStrategy]

findWeakness [AssignmentID] [StudentID] [RankingStrategy]

note

We will guarantee the “student” command contains at least one [StudentID].

We will guarantee the “schoolStrategy” command contains at least one [Level],[Score]

We will guarantee “student” and “schoolStrategy” command be given before other command.

“schoolStrategy” will only appear once in the same test case.

note

There will only be one successful “assignment” command for an individual assignment.

note

[RankingStrategy] is limited to "MeanRankingStrategy" and "MedianRankingStrategy".

Input File:

RubricFile format

[Criterion] [Level] [Description]

[Criterion] [Level] [Description]

[Criterion] [Level] [Description]

...

note

[Criterion], [Level] and [Description] in RubricFile are split by “tab”.

There is no space in [Criterion] and [Level].

There may contain space in [Description].

[Level] will be guaranteed in the current schoolStrategy.

The same [Criterion] will be in consecutive rows.

The same [Criterion], [Level] order is the same as schoolStrategy.

SampleScoreFile format

[Level]

[Level]

[Level]

...

note

[Level] is linguistic word.

[Level] will be guaranteed in the current schoolStrategy.

You should read this file to rank other students’ assignment.

You only have to read the top k [Level] for ranking. (k = number of criterion in the assignment)

Output:

After assignment command:

```
If ([Reviewer] and [Student] are the same) // fail
    => Cannot review one's own assignment.
Else If (number of reviewers < 3 or number of reviewers > 5) // fail
    => Assignment should be reviewed by 3-5 students.
Else
    // For each reviewer, peer review by reading [SampleScoreFile], then print the following format.
    // The number of printed [Level] is equal to the number of assignment criterion.
    => [AssignmentID]-[StudentID] was reviewed by [ReviewerID]. Level: [Level] [Level] ...
```

After printRubric command:

```
// Print the specific assignment rubric in the following format
// The order is the same as [RubricFile]
=> ([Criterion],[Level]) [Description]
=> ...
```

After averageCriterion command:

```
// Find all assignments by [AssignmentID], for each criterion
// Print average (sum all scores associated with this criterion, then divide) score in the following format
// AvgScore should round to the first decimal place
=> Assignment: [AssignmentID], Criterion: [Criterion], AvgScore: [AvgScore]
=> ...
```

After calculateScore command:

```
// For each criterion, calculate score by [RankingStrategy]
// Average all criterion ranking scores to final score, then print final score in the following format
// Final score should round to the first decimal place
=> Assignment: [AssignmentID], Student: [StudentID], Score: [FinalScore]
```

After findStrength command:

```
// For each criterion, calculate score by [RankingStrategy]
// Find the criterion with the highest score, then print in the following format
// If multiple criteria have the same score, then print in input order (RubricFile)
=> Assignment: [AssignmentID], Student: [StudentID], Strength: [Criterion] [Criterion] ...
```

After findWeakness command:

```
// For each criterion, calculate score by [RankingStrategy]
// Find the criterion with the lowest score, then print in the following format
// If multiple criteria have the same score, then print in input order (RubricFile)
=> Assignment: [AssignmentID], Student: [StudentID], Weakness: [Criterion] [Criterion] ...
```

Comment:

Please implement your main function in Class Main.
We'll test your program through "java Main inputFile"
e.g java Main sampleInput

Do not read input from System.in or hard code input file, or your program won't pass any test case.

Upload:

Please zip your source code and upload it.
The file name should be Team[teamID].zip. e.g. Team7.zip
The folder structure should be:
 unzip Team7.zip
=> [dir] Team7
=> Team7/Main.java
=> Team7/*.java (optional)

You won't receive any point if you didn't follow the directory structure or main class name or compressed format!

Sample criteria1.txt:

Knowledge/Understanding	Excellent	The presentation demonstrates a depth ...
Knowledge/Understanding	Competent	The presentation uses knowledge which ...
Knowledge/Understanding	NeedsWork	The presentation uses little relevant ...
Thinking/Inquiry	Excellent	The presentation is centered around ...
Thinking/Inquiry	Competent	The presentation shows an analytical ...
Thinking/Inquiry	NeedsWork	The presentation shows no analytical ...
Communication	Excellent	The presentation is imaginative and ...
Communication	Competent	Presentation techniques used are effective ...
Communication	NeedsWork	The presentation fails to capture ...
UseOfVisualAids	Excellent	The presentation includes appropriate and ...
UseOfVisualAids	Competent	The presentation includes appropriate visual ...
UseOfVisualAids	NeedsWork	The presentation includes no visual ...

Sample score1.txt:

Excellent
Excellent
Competent
Competent
NeedsWork
NeedsWork

Sample score2.txt:

Competent
Competent
Competent
Competent
Competent
Competent

Sample Input:

```
student 001 002 003 004
schoolStrategy Excellent,3 Competent,2 NeedsWork,1
designCriterion A1 criteria1_1.txt
assignment A1 001 002,score1_1.txt
assignment A1 001 002,score1_1.txt 003,score1_1.txt 004,score1_2.txt
assignment A1 002 001,score1_1.txt 003,score1_1.txt 004,score1_2.txt
assignment A1 003 001,score1_1.txt 002,score1_1.txt 004,score1_2.txt
assignment A1 004 001,score1_1.txt 002,score1_1.txt 003,score1_2.txt
printRubric A1
averageCriterion A1
calculateScore A1 001 MeanRankingStrategy
findStrength A1 002 MeanRankingStrategy
findWeakness A1 003 MeanRankingStrategy
```

Sample Output:

Assignment should be reviewed by 3-5 students.

A1-001 was reviewed by 002. Level: Excellent Excellent Competent Competent

A1-001 was reviewed by 003. Level: Excellent Excellent Competent Competent

A1-001 was reviewed by 004. Level: Competent Competent Competent Competent

A1-002 was reviewed by 001. Level: Excellent Excellent Competent Competent

A1-002 was reviewed by 003. Level: Excellent Excellent Competent Competent

A1-002 was reviewed by 004. Level: Competent Competent Competent Competent

A1-003 was reviewed by 001. Level: Excellent Excellent Competent Competent

A1-003 was reviewed by 002. Level: Excellent Excellent Competent Competent

A1-003 was reviewed by 004. Level: Competent Competent Competent Competent

A1-004 was reviewed by 001. Level: Excellent Excellent Competent Competent

A1-004 was reviewed by 002. Level: Excellent Excellent Competent Competent

A1-004 was reviewed by 003. Level: Competent Competent Competent Competent

(Knowledge/Understanding,Excellent) The presentation demonstrates a depth ...

(Knowledge/Understanding,Competent) The presentation uses knowledge which ...

(Knowledge/Understanding,NeedsWork) The presentation uses little relevant ...

(Thinking/Inquiry,Excellent) The presentation is centered around ...

(Thinking/Inquiry,Competent) The presentation shows an analytical ...

(Thinking/Inquiry,NeedsWork) The presentation shows no analytical ...

(Communication,Excellent) The presentation is imaginative and ...

(Communication,Competent) Presentation techniques used are effective ...

(Communication,NeedsWork) The presentation fails to capture ...

(UseOfVisualAids,Excellent) The presentation includes appropriate and ...

(UseOfVisualAids,Competent) The presentation includes appropriate visual ...

(UseOfVisualAids,NeedsWork) The presentation includes no visual ...

Assignment: A1, Criterion: Knowledge/Understanding, AvgScore: 2.7

Assignment: A1, Criterion: Thinking/Inquiry, AvgScore: 2.7

Assignment: A1, Criterion: Communication, AvgScore: 2.0

Assignment: A1, Criterion: UseOfVisualAids, AvgScore: 2.0

Assignment: A1, Student: 001, Score: 2.3

Assignment: A1, Student: 002, Strength: Knowledge/Understanding Thinking/Inquiry

Assignment: A1, Student: 003, Weakness: Communication UseOfVisualAids