## Problem 1 – Exam Results ???

You are given **N** lines with exam results in the following format:

**[Student] [Course] [Exam points] [Course bonuses]**

Your task is to **calculate** the **course points** and **the gradе (3.00 ... 6.00)** for each student. **Exam points are 80% per course point**. **Example:** *400 exam points = 80 course points, 200 = 40*. If **exam points are less than 100** the result is **failed**.

After you calculate the exam points to course points, you must **add the bonus points**. Now **calculate the grade**. **Minimum points for 6.00 are 80**, so the formula is: **((course point / needed points for 6.00(80points)) \* 4) + 2**.

After the results you are given also a line with a **course name**. For **all exams from this course** you must **find the average exam points**.

### Input

* The input is passed to the first JavaScript function found in your code as an **array of N+1 strings**.
* The first **N** strings hold the results, separated by one or more whitespaces.
* The last string shows the course name for which average points are requested.

### Output

* The output should consist of **N+1** lines.
  + **“[student]**: Exam – **"[course name]"**; Points – **[course point]**; Grade – **[grade]**”
    - Points must be rounded to two digits after the decimal point. Trailing zeroes should be removed. For example, 85.11 is displayed as 85.11, 49.30 is displayed as 49.3, and 80.00 is displayed as 80.
    - Grades must be rounded to two digits after the decimal point. Trailing zeroes stay.
  + “**[Student]** failed at **"[course name]"**” - If exam points are less than 100 points.
* The last line is the average point from the requested exam

“"**[Course name]**" average points -> **[avg. point]**”

* + Points must be rounded to two digits after the decimal point. Trailing zeroes should be removed.

### Constraints

* **Lines with results** will be in the range **[1 .. 100]**
* **Exam points** will be in the range **[0 .. 400]**
* Allowed working time/memory **0.15s/16MB**

**function** *solve*(input) {  
  
 **let** requestedCourse = input.pop();  
 **let** averagePoints = 0;  
 **let** count = 0;  
  
 **for** (**let** i = 0; i < input.length; i++) {  
  
 **let** regex = /(.+?)\s+(.+?)\s+([0-9]+)\s+([0-9]+)/g;  
 **let** mach = regex.exec(input[i]);  
  
 **let** nameMach = mach[1];  
 **let** courseMach = mach[2];  
 **let** examPointsMach = mach[3];  
 **let** bonusMach = mach[4];  
  
 **if**(courseMach == requestedCourse){  
  
 averagePoints += Number(examPointsMach);  
 count++;  
 }  
  
 **if**(examPointsMach < 100){  
 **console**.log(**`**${nameMach} **failed at "**${courseMach}**"`**)  
 }  
 **else**{  
 **let** points = (examPointsMach - (examPointsMach\*80/100)) + Number(bonusMach);  
 points = (points\*100)/100;  
  
 **let** grade = ((points/80)\*4)+2;  
  
 **if**(grade > 6){  
 grade = 6;  
 }  
  
 **console**.log(**`**${nameMach}**: Exam – "**${courseMach}**"; Points –** ${points}**; Grade –** ${grade.toFixed(2)}**`**)  
 }  
 }  
 **console**.log(**`"**${requestedCourse}**" average points ->** ${averagePoints/count}**`**)  
  
  
}  
  
*solve*([  
 **'Pesho C#-Advanced 100 3'**,  
 **'Gosho Java-Basics 157 3'**,  
 **'Tosho HTML&CSS 317 12'**,  
 **'Minka C#-Advanced 57 15'**,  
 **'Stanka C#-Advanced 157 15'**,  
 **'Kircho C#-Advanced 300 0'**,  
 **'Niki C#-Advanced 400 10'**,  
 **'C#-Advanced'**]);

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Pesho C#-Advanced 100 3  Gosho Java-Basics 157 3  Tosho HTML&CSS 317 12  Minka C#-Advanced 57 15  Stanka C#-Advanced 157 15  Kircho C#-Advanced 300 0  Niki C#-Advanced 400 10  C#-Advanced | Pesho: Exam – "C#-Advanced"; Points – 23; Grade – 3.15  Gosho: Exam – "Java-Basics"; Points – 34.4; Grade – 3.72  Tosho: Exam – "HTML&CSS"; Points – 75.4; Grade – 5.77  Minka failed at "C#-Advanced"  Stanka: Exam – "C#-Advanced"; Points – 46.4; Grade – 4.32  Kircho: Exam – "C#-Advanced"; Points – 60; Grade – 5.00  Niki: Exam – "C#-Advanced"; Points – 90; Grade – 6.00  "C#-Advanced" average points -> 202.8 |

**function** *solve*(input) {  
  
 **let** requestedCourse = input[input.length - 1].trim();  
 **let** averagePoints = 0;  
 **let** count = 0;  
  
 **for** (**let** i = 0; i < input.length -1; i++) {  
  
 **let** currentLine = input[i].**split**(/\s+/).filter(**function** (x) {**return** x});  
 **let** nameMach = currentLine[0];  
 **let** courseMach = currentLine[1];  
 **let** examPointsMach = currentLine[2];  
 **let** bonusMach = currentLine[3];  
  
 **if**(courseMach == requestedCourse){  
  
 averagePoints += Number(examPointsMach);  
 count++;  
 }  
  
 **if**(examPointsMach < 100){  
 **console**.log(**`**${nameMach} **failed at "**${courseMach}**"`**)  
 }  
 **else**{  
 **let** points = (examPointsMach - (examPointsMach\*80/100)) + Number(bonusMach);  
 points = (points\*100)/100;  
  
 **let** grade = ((points/80)\*4)+2;  
  
 **if**(grade > 6){  
 grade = 6;  
 }  
  
 **console**.log(**`**${nameMach}**: Exam - "**${courseMach}**"; Points -** ${points}**; Grade -** ${grade.toFixed(2)}**`**)  
 }  
  
 }  
 **let** sum = averagePoints/count;  
 **console**.log(**`"**${requestedCourse}**" average points ->** ${(sum\*100)/100}**`**);  
  
  
}  
  
*solve*([  
 **'Pesho C#-Advanced 100 3'**,  
 **'Gosho Java-Basics 157 3'**,  
 **'Tosho HTML&CSS 317 12'**,  
 **'Minka C#-Advanced 57 15'**,  
 **'Stanka C#-Advanced 157 15'**,  
 **'Kircho C#-Advanced 300 0'**,  
 **'Niki C#-Advanced 400 10'**,  
 **'C#-Advanced'**]);