

Name: _____ Section _____

Exam2. Spring18. Liron

Keep track of the time. Everyone's different. Log in now. If you are already locked out, go get your charger for your phone now. TESTING EXPECTED AS SOON AS YOU CAN: trap for each while loop. Once for full run of code.

ONLY USE CONTENT OF CLASS TAUGHT UP TO TODAY. NO CREDIT OTHERWISE.

Do not miss submission: No exceptions. No emails. Nothing. Zero.

I did my best to not make mistakes on this cover sheet. If you think there is one, be an engineer and fix it - but do not change the requirements. I will not answer questions during the first hour - I believe in you, you can do this! ☺ Show what you can do, hardcode where you get stuck!

It's snowboarding "slopestyle" competition at the Winter Olympics! In "slopestyle", riders perform tricks while descending a course. The course is full of obstacles including boxes, rails, jumps, etc... Sadly, the rider may stumble or fall during each trick and loses points!



After a quick welcome, your code prompts the rider how many tricks they did. This value needs to be a value greater than 3. Trap the user when empty or invalid values are entered. (Check screenshot).

```
** Snowboarding Slopestyle at the olympics!!! **  
  
How many tricks did you do (at least 4)? 3  
CAUTION: whole>=4 only: 5.5  
CAUTION: whole>=4 only:  
CAUTION: whole>=4 only: 4
```

Your code will then proceed to show/ask data for each trick: some user input, some calculated, some randomized!

Some requirements/specifications:

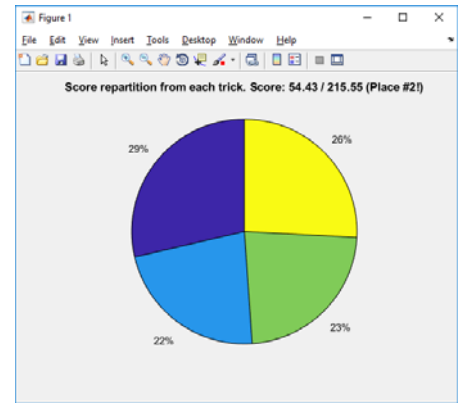
- 1) Display the trick's number each time: #1, #2, #3, followed by the type of trick. The type of trick is randomized each time:
 - a. Create a database with the three types: 'box', 'jump', 'rail' - in that order.
 - b. Generate a random integer 1 to 3: this will be the index (i.e. position) of the word to show.
 - c. Display - **NO if statement**. (1LOC)
- 2) Riders are scored on how smooth each trick was. Prompt on a scale from 1 to 5.
 - a. You must trap when it is invalid (see screenshot for trick #1). This value must be a whole value 1 to 5.
- 3) The maximum score for a trick is generated each trick: a random decimal between 45 and 65.
 - a. Using a running total **only**, keep adding those full points for the end.
 - b. To calculate the score based on the smoothness, use: $\text{actualScore} = \text{maxScore} * (0.5 + 0.1 * \text{smoothness})$.
 - c. Display both values with 2 decimals.
- 4) Last, if the trick is a jump (see figure #2 and #3 for example) and the rider stays in the air greater than 3seconds (random integer 1 to 4 using rand() and a rounding function of your choice), then the rider gets a bonus of 10points! Display number of seconds and bonus, as shown above.

```
Trick #1 (a rail!)  
How smooth did that trick go? 1-5(smoothest) : 0  
ERROR: whole 1-5(smoothest) : 6  
ERROR: whole 1-5(smoothest) : 4.4  
ERROR: whole 1-5(smoothest) :  
ERROR: whole 1-5(smoothest) : 5  
--> so 60.34 / 60.34 so far!  
** This trick: 60.34 / 60.34  
** Totals: 60.34 / 60.34  
  
Trick #2 (a jump!)  
How smooth did that trick go? 1-5(smoothest) : 4  
--> so 47.31 / 52.57 so far!  
You only stayed 3 seconds in the air! (Bonus 0)  
** This trick: 47.31 / 52.57  
** Totals: 107.66 / 112.91  
  
Trick #3 (a jump!)  
How smooth did that trick go? 1-5(smoothest) : 3  
--> so 38.57 / 48.22 so far!  
You stayed 4 seconds in the air, yay (Bonus +10)|  
** This trick: 48.57 / 48.22  
** Totals: 156.23 / 161.13  
  
Trick #4 (a boxe!)  
How smooth did that trick go? 1-5(smoothest) : 5
```

- 5) Calculate as you wish the total score for the rider. The summary on the screenshot `**Totals: XXX/XXX` is to help you reverse engineer but is not necessary to show.
- 6) Store the score of each trick in a vector, so you can generate a pie graph later.

Last, use `pie()`; to generate the vertical bar graph. This requires one parameter: your vector from 6). Do not store the return value. (note: the percent and colors are automatically calculated by MATLAB).

- a. The title must be dynamic: showing the actual score `xxx/xxx`.
- b. For extra credit, generate a random integer 1 to 6. This is their place! Also show in the dynamic title.



And you're done! Save, check the rubric peacefully. ZIP and submit before the XX:55am/pm deadline!

Rubric:

- | | |
|--------|-----------------------------------------------------------------------------------|
| 3 | Name, section, description |
| 5 | Clean ups |
| 10 | Full descriptive algorithm - can be done with no knowledge of programming |
| 5 | Valid/descriptive variable names (use <code>which</code> to check) |
| 5 | Spacing of code - consistent |
| 5 | Spacing of output (human factor aspect) |
| 5 | Indent of code according to conventions |
| 5 | All proper semicolons. Use <code>fprintf()</code> for any output. |
| 10 | Code must not crash - comment what crashes, hardcode if necessary some values |
| 3 | Welcome message |
| 5, 15 | Prompt number of tricks, trap invalid |
| 10 | Loop to ask data for each trick |
| 5 | Display trick's counter |
| 10 | Database for trick type |
| 5 | Generate random index for trick |
| 10 | Display type of trick - NO if statement |
| 5, 20 | Prompt trick's smoothness, trap invalid |
| 5 | Random maximum score |
| 10 | Keep running total |
| 5 | Calculate actual score |
| 5 | Generate random time in the air |
| 15 | Determine bonus points if a jump |
| 5 | Display bonus for jump |
| 5 | Calculate actual score for that trick |
| 5 | Store in vector score for that trick |
| 5 | Create pie graph |
| 10 | Dynamic title |
| +5 | random rank for title |
| 10, 10 | TEST: each loop must show it traps properly (do not wait for code to be finished) |
| 10 | TEST: one for overall code - or crashes if it crash. |