

Comments on Test 2B & How to Estimate Your Course Grade

Below are our comments from reading your work on Test 2B. Be sure to try to re-do any Test 2B question on which you didn't do well and approach any member of the course staff for help. Do not just read the solutions! (which will be posted later this week)

Regrade Requests

If you see a *grading error*--the grader mistook your correct code to be incorrect and deducted points--then please submit a regrade request on Gradescope and specify exactly what and where the grading error is. On the other hand, do not ask for a regrade if you dislike the number of points deducted--the same grading rubric was applied for all students and will not be changed.

You can submit a Test 2B regrade request on Gradescope until 9am on Monday, 05/18.

Question 1 (team up)

Students did well:

- Students who used the built-in `sort()` function generally did so correctly and made proper use of the permutation indices.
- Good use of `rem()` to split into teams. There are other ways to do this as well (see lecture examples on dealing hands of cards), but `rem()` is familiar and easy to generalize. This resulted in team assignments (in increasing order of height) generally being correct.
- Very few out-of-bounds errors even when the number of students wasn't divisible by 3 - good job learning from Test 2A!

Could use improvement:

- The output was supposed to contain handles to the same `Student` objects that were passed in, but many students instead added the names (or sometimes heights) into the team array.
- The syntax used to assign elements to the output cell array was pretty inconsistent. Single-element access should use curly braces (with a comma between row and column indices). Subarray access, however, needs to use parentheses (this includes vectorized reordering using the permutation indices). And if curly braces are used on the LHS, but parentheses on the RHS, that actually creates nested cells.
- When accessing properties of the `students` cell array, several people attempted to do "bulk access" with an expression like `students.height`, instead of accessing students one-by-one. The former does not work for cell arrays. Similarly, `sort()` does not work on general cell arrays; the heights must be extracted and stored in a simple array to sort them.
- Some students attempted to sort by height manually. This made the problem unnecessarily difficult, and most implementations were incorrect. Be aware of which built-in functions are available and how to use them.

Question 2 (update balance)

Students did well:

- Students did a good job of creating distinct fid's for both files (not overwriting each other). Students also did a good job opening and closing both files, and iterating through them until `feof()`.
- Many students did well in checking the Deposit and Withdraw conditions and updating the `minbal` and `fbal` accordingly.
- Students did a good job implementing the best-so-far pattern to find the minimum balance.

Could use improvement:

- Many students converted both of the files into cell arrays, which ends up taking a lot of time and memory. Instead, students could iterate through 'Balances-Mar.txt' just until the correct ID number, and then loop through the entirety of 'Statement-Apr.txt' once, updating the `fbal` and `minbal` as they go.
- A handful of students used functions not allowed on the exam, especially `str2cell()`. Many other students wrote their own tokenization code for each line. Note that, for fixed columns, there's no need to search for spaces to split the string. It is much easier to use the indices provided in the problem statement and recognize that `str2double()` will ignore empty spaces in a char array.
- Some students used the first letter of the lines to determine a DEPOSIT or WITHDRAW instead of doing a `strcmp` of the entire word. For instance, if the first letter is a D, the students assumed this is a Deposit. The problem said that other lines must be ignored, so this is an unsafe assumption. Similarly, some students used an 'if' statement checking for the term 'DEPOSIT', but just used 'else' when checking for 'WITHDRAW'. This additionally causes issues in the second (header) line of the file, which does not have any transaction numbers in it.
- Some students returned arrays rather than single values for `fbal` and `minbal`.

Question 3 (Vert class)

Students did well:

- Many students got both parts correct. This suggests a good understanding of how objects are stored and passed by reference (even to their own methods).

Could use improvement:

- For part (a), a few students tried to return an output from the `scale()` method, when none was expected.
- A few students forgot to access properties through the self reference when assigning the scaled properties.
- For part (b), the majority of the mistakes were in formatting: some printed variable names in front of the numeric values, and some printed all numbers on the same line. Note that `disp(variable)` only prints the value stored in that variable (in contrast to leaving off a semicolon), each within its own single line.

Statistics

Max: 100% (50/50)
Median: 92%
Mean: 86.7%
S.Dev: 15%

% Action to take after prelim

```
if yourScore > 43/50

    toDo= celebrate + later look at the solution

elseif yourScore > 32/50

    toDo= redo problems + later check solution

else

    toDo= meet with course staff to go over exam ...
        + stay caught up from now on ...
        + do NOT just read the solution--must redo problems

end
```

How to estimate your course grade

All work through Test 2B (including Project 5, Exercise 12, and in-lecture quizzes) has had their scores uploaded to CMS. Additionally, each assignment has been weighted as described in the syllabus to yield a “total weighted score” up to this point. Here is how to use this total score to estimate your standing in the course:

1. If you submitted Exercise 13, add 0.3 to your total.
2. Estimate your Project 6 grade as a number between 0 and 1 (for example, average your previous project scores and divide by 10). Multiply this by 7.5 and add it to your total.
3. To account for the Final, there are two ways you can look at it:
 - a. Estimate your Final grade as a number between 0 and 1 (for example, by averaging your previous test scores and normalizing; add a 1% bonus if you plan to submit a course evaluation). Multiply this by 24 and add it to your total. Compare with the grade ranges on the syllabus to estimate your letter grade.
 - b. Alternatively, take the grade you hope to achieve in the course out of 100 (so 65 for an S, for example). Subtract your total, then divide by 24. This tells you the score you need to achieve on the Final (as a percentage between 0 and 1) in order to earn that grade.

Example 1

- Suppose your total weighted score in CMS is 59.7
- You submitted Exercise 13, so add 0.3, bringing your total to 60
- Your average project score was 8/10, so multiply $0.8 \times 7.5 = 6$ and add it to your total, yielding 66
- Your average test score so far is 85%, so multiply $0.85 \times 24 = 20.4$ and add it to your total, yielding 86.4. Looking at the syllabus, you would likely earn a “B” if you maintained your performance.

Example 2

- Suppose your total weighted score in CMS is 40
- You neglected to submit Exercise 13, so your total is still 40
- Your average project score was 7/10, so multiply $0.7 \times 7.5 = 5.25$, bringing your total to 45.25
- You would like to earn an S (C-) in the course, which corresponds to a 65. Compute $(65 - 45.25)/24 = 0.82$. Therefore, you need to score an 82% on the Final in order to earn an S.

Note that in many colleges, a “D” (>55) is considered a “marginal pass,” so unless you are very confident about the Final, S/U may not be the best choice for you. We recommend you reach out to your academic advisor.

Note that this estimate is *conservative*. It does not account for dropping your lowest project grade (if above 5/10), nor does it account for replacing your Prelim 1 grade (if you scored better on subsequent tests). It will also not reflect any special circumstances that you may have discussed with the instructor (absences due to extended illness, for example). Therefore, your actual final grade may be slightly higher. This effect is probably smaller than the uncertainties surrounding your P6 and Final grades, however.

Finally, just for clarity, the “lecture questions” assignment in CMS reflects all in-lecture quiz problems: iClicker questions, Canvas quizzes, and the Gradescope practice submission. The maximum score of 36 is $\frac{1}{2}$ the maximum number of possible points. This score will be updated during Finals week with the last few Canvas questions. If your score is much lower than you expect, it probably means that your iClicker was not successfully registered; register it in Canvas and contact Dr. Muhlberger.