

CSC 110

Introduction to Computer Science

Homework Assignment 3

Conditionals

Due: By Lecture 14, midnight

Note: Homework Assignment 3 should be completed individually.

Objective

So far, we have trusted that the user will input the desired type of values when we request them to do so.

We will now do some basic input checking to see if the input that the user provided is adequate for the task.

The theme of this homework is a “grade” calculator.

For this homework you need to complete three tasks:

- 1) Make the `read_five_ints` function with input checking for each requested number.
- 2) Make the `pick_averaging_method` function with input checking for the requested option.
- 3) Make the `pick_visualization` function with input checking for the requested option.

So you understand what we are shooting for, an example run looks like this:

```
Give me the next grade in [0 to 10]:8
Give me the next grade in [0 to 10]:6
Give me the next grade in [0 to 10]:10
Give me the next grade in [0 to 10]:7
Give me the next grade in [0 to 10]:6
Sorted grades: [6, 6, 7, 8, 10]
Pick 'a' for mean, 'b' for median, 'c' for mode: b
picked: Median
Pick '1' for print average, or '2' for plot average: 2
Annotated grades:
 6  6 (7)8 10
The End
```

You only need to complete the functions. The `main` is provided for you and it takes care of all the printouts except the error messages we require you to print (more on that below). In particular, it takes care of sorting the input grades and printing the desired formatted results. **We only care about detecting mistakes in the user input choices.**

The DocStrings below every function explain what they do. Do not remove them.

A Note on passing the tests

Any tests are very strict with respect to the format of requested prompts and printouts so pay attention to exactly what is requested and replicate it as exactly as possible.

The global variable `grades`

The list variable `grades` is declared at the top and outside any function. This makes it a member of the **global context** and visible to all functions.

- This variable is initialized with 5 zeros inside.
- If you modify this list while inside any function, the global variable will be modified. However, you *need* to tell python that you intend to use the **global** `grades` so make sure you put that at the top of the function.
- You can overwrite a value at some index `idx` with the number `num` by using the command: `grades[idx] = num`

- remember that the only valid indices that you can use for the variable `grades` are 0 through 4 (because we only have a total of 5).

Task 1: read_five_ints

define the function

The objective of this function is to read five inputs from the user, cast them to integers, and use them to update the list.

Tip: To check if an input string (let's call it `in_str`) is composed of digits, you can use this notation:

`in_str.isdigit()`

- This returns `True` if `in_str` is composed of only digits e.g. "123" or "0107".
- This returns `False` if `in_str` is not composed of only digits e.g. "123.7" (note the dot) or "01 7" (note the space), or "02q5".

The steps to complete the function are detailed below:

1. Remove the `pass` keyword inside the provided for loop and fill it out with your code.
2. You should use the `input` function to read one number using the following prompt: "Give me the next grade in [0 to 10]:"; Note that whatever is read is a String.
3. Check to see if the input is only digits. If it is not, print an error message with: "Error in read_five_ints: input string is not for an integer" and follow that with the statement `exit()`.
4. If the string has only digits, cast the input string into an integer by using the `int` function.
5. If the resulting integer is not inside the interval [0,10] (between 0 and 10 inclusive), you should print an error message with: "Error in read_five_ints: input integer outside of range" and follow that with the statement `exit()`.
6. If the integer is in the correct interval, overwrite the grade at index `idx` with the integer read from the user.
7. After the loop is done, the grades should only have the 5 integers provided by the user.
8. Remember, after modifying `grades` for all indices, you are done, you don't need to return anything.

Testing

This is a good point to run the tests and check to see if this function works as indicated (task1 tests), for both correct inputs or incorrect ones (two types of incorrect inputs).

Note that for the tests, we do not compare against the prompts and user input, only the printouts.

Task 2: pick_averaging_method

define the function

The purpose of `pick_averaging_method` is to pick one of three standard averaging methods: mean, median, or mode. Mean is the standard arithmetic average we are used to, e.g. the mean of [6,6,7,8,10] is 7.4; median is the value at the center of the provided values if they are arranged in a sorted list, e.g. the median of [6,6,7,8,10] is 7, and the median of [6,6,7,8] is 6.5 (in the middle between 6 and 7); Lastly, the mode is the most repeated value, e.g. the mode of [6,6,7,8,10] is 6, and the mode of [6,6,7,8,8] is 6 (in case of a tie, it's the first one);

You need to do the following:

1. remove the `pass` keyword and instead:
2. get a string input from the user, using the prompt "Pick 'a' for mean, 'b' for median, 'c' for mode: ".
3. Use a multi-path conditional to decide one of the following options:
 1. if the user input "a", print the message


```
picked: Mean
```

 then calculate the mean by using the command:

```

    avg = statistics.mean(grades)
    and return the obtained value.
2. if the user input "b", print the message
    picked: Median
    then calculate the mean by using the command:
    avg = statistics.median(grades)
    and return the obtained value.
3. if the user input "c", print the message
    picked: Mode
    then calculate the mean by using the command: avg = statistics.mode(grades) and return the obtained
    value.
4. Otherwise, print an error message with: "Error in pick_averaging_method: incorrect option
    picked" and follow that with the statement exit().

```

Testing

This is a good point to run the tests and check to see if this function works as indicated, for both correct inputs or incorrect ones (only one type of error here).

Task 3: pick_visualization

define the function

The purpose of pick_visualization is to print the average result in one of two ways, depending on user input. If the user picks option 1, the simple average is printed after the list, and if option 2 is picked, a “fancy” version of the list items is printed out, with annotations to indicate where the average is located.

You need to do the following:

1. remove the `pass` keyword and instead:
2. get a string input from the user, using the prompt "Pick '1' for print average, or '2' for plot average: ".
3. if the user input "1", you should simply call the provided function `print_list_and_average` with the call `print_list_and_average(average)`. This will take care of printing the message (placeholders in use):
The average <[list printed here]> is <average>
see the examples shown below.
4. if the user input "2", you should simply call the provided function `plot_grades` with the call: `plot_grades(average)`.
5. Otherwise, print an error message with: "Error in pick_visualization: incorrect option picked" and follow that with the statement `exit()`.

Testing

This is a good point to run the tests and check to see if this function works as indicated, for both correct inputs or incorrect ones (only one type of error here).

Output Examples

Correct Input

A: Correct input, avg: mean, normal printout

Example: if the inputs the user gives are 6, 6, 7, 8, and 10, then the `grades` list should contain [6, 6, 7, 8, 10]; If then, the user selects option "a", then the average should be 7.4; If then, the user selects option "1", the complete Console information shown would be this:

```
Give me the next grade in [0 to 10]:6
Give me the next grade in [0 to 10]:6
Give me the next grade in [0 to 10]:7
Give me the next grade in [0 to 10]:8
Give me the next grade in [0 to 10]:10
Sorted grades: [6, 6, 7, 8, 10]
Pick 'a' for mean, 'b' for median, 'c' for mode: a
picked: Mean
Pick '1' for print average, or '2' for plot average: 1
The average of [6, 6, 7, 8, 10] is 7.4
The End
```

B: Correct input, avg: median, normal printout

Example: if the inputs the user gives are 6, 6, 7, 8, and 10, then the `grades` list should contain [6, 6, 7, 8, 10]; If then, the user selects option "b", then the average should be 7; If then, the user selects option "1", the complete Console information shown would be this:

```
Give me the next grade in [0 to 10]:6
Give me the next grade in [0 to 10]:6
Give me the next grade in [0 to 10]:7
Give me the next grade in [0 to 10]:8
Give me the next grade in [0 to 10]:10
Sorted grades: [6, 6, 7, 8, 10]
Pick 'a' for mean, 'b' for median, 'c' for mode: b
picked: Median
Pick '1' for print average, or '2' for plot average: 1
The average of [6, 6, 7, 8, 10] is 7
The End
```

C: Correct input, avg: mode, normal printout

Example: if the inputs the user gives are 6, 6, 7, 8, and 10, then the `grades` list should contain [6, 6, 7, 8, 10]; If then, the user selects option "c", then the average should be 7; If then, the user selects option "1", the complete Console information shown would be this:

```
Give me the next grade in [0 to 10]:6
Give me the next grade in [0 to 10]:6
Give me the next grade in [0 to 10]:7
Give me the next grade in [0 to 10]:8
Give me the next grade in [0 to 10]:10
Sorted grades: [6, 6, 7, 8, 10]
Pick 'a' for mean, 'b' for median, 'c' for mode: c
picked: Mode
Pick '1' for print average, or '2' for plot average: 1
The average of [6, 6, 7, 8, 10] is 6
The End
```

D: Correct input, avg: mean, fancy printout

Example: if the inputs the user gives are 6, 6, 7, 8, and 10, then the `grades` list should contain [6, 6, 7, 8, 10]; If then, the user selects option "a", then the average should be 7; If then, the user selects option "2", the complete Console information shown would be this:

```

Give me the next grade in [0 to 10]:6
Give me the next grade in [0 to 10]:6
Give me the next grade in [0 to 10]:7
Give me the next grade in [0 to 10]:8
Give me the next grade in [0 to 10]:10
Sorted grades: [6, 6, 7, 8, 10]
Pick 'a' for mean, 'b' for median, 'c' for mode: a
picked: Mean
Pick '1' for print average, or '2' for plot average: 2
Annotated grades:
 6 6 7 ^8 10
The End

```

Note that the ^ mark is an annotation of where the average is located between the grades. You do not need to do this, it is provided.

E: Correct input, avg: median, fancy printout

Example: if the inputs the user gives are 6, 6, 7, 8, and 10, then the `grades` list should contain [6, 6, 7, 8, 10]; If then, the user selects option "b", then the average should be 7; If then, the user selects option "2", the complete Console information shown would be this:

```

Give me the next grade in [0 to 10]:6
Give me the next grade in [0 to 10]:6
Give me the next grade in [0 to 10]:7
Give me the next grade in [0 to 10]:8
Give me the next grade in [0 to 10]:10
Sorted grades: [6, 6, 7, 8, 10]
Pick 'a' for mean, 'b' for median, 'c' for mode: b
picked: Median
Pick '1' for print average, or '2' for plot average: 2
Annotated grades:
 6 6 (7)8 10
The End

```

Note that the parentheses are an annotation that marks which grade is the median. You do not need to do this, it is provided.

F: Correct input, avg: mode, fancy printout

Example: if the inputs the user gives are 6, 6, 7, 8, and 10, then the `grades` list should contain [6, 6, 7, 8, 10]; If then, the user selects option "c", then the average should be 7; If then, the user selects option "2", the complete Console information shown would be this:

```

Give me the next grade in [0 to 10]:6
Give me the next grade in [0 to 10]:6
Give me the next grade in [0 to 10]:7
Give me the next grade in [0 to 10]:8
Give me the next grade in [0 to 10]:10
Sorted grades: [6, 6, 7, 8, 10]
Pick 'a' for mean, 'b' for median, 'c' for mode: c
picked: Mode
Pick '1' for print average, or '2' for plot average: 2
Annotated grades:
(6)(6)7 8 10
The End

```

Note that the parentheses are an annotation that marks which grades are the mode. You do not need to do this, it is provided.

Incorrect input

G: Incorrect number entry (second entry is not only digits)

Example: if the inputs the user gives are 6, and pizza, then an error is detected and a printout is made. The complete Console information shown would be this:

```
Give me the next grade in [0 to 10]:6
Give me the next grade in [0 to 10]:pizza
Error in read_five_ints: input string is not for an integer
repl process died unexpectedly:
```

Note that the message “repl process died unexpectedly:” is particular to replit when the exit function is used.

H: Incorrect number entry (second number is not in [0,10])

Example: if the inputs the user gives are 6, and 17, then an error is detected and a printout is made. The complete Console information shown would be this:

```
Give me the next grade in [0 to 10]:6
Give me the next grade in [0 to 10]:17
Error in read_five_ints: input integer outside of range
repl process died unexpectedly:
```

Note that the message “repl process died unexpectedly:” is particular to replit when the exit function is used.

I: Incorrect average option

Example: if 5 valid numbers are given, and the input for the average option the user gives is d, then an error is detected and a printout is made. The complete Console information shown would be this:

```
Give me the next grade in [0 to 10]:6
Give me the next grade in [0 to 10]:6
Give me the next grade in [0 to 10]:7
Give me the next grade in [0 to 10]:8
Give me the next grade in [0 to 10]:10
Sorted grades: [6, 6, 7, 8, 10]
Pick 'a' for mean, 'b' for median, 'c' for mode: d
Error in pick_averaging_method: incorrect option picked
repl process died unexpectedly:
```

Note that the message “repl process died unexpectedly:” is particular to replit when the exit function is used.

J: Incorrect visualization option

Example: if 5 valid numbers are given, a correct average option is chosen, but the input for the visualization option the user gives is 3, then an error is detected and a printout is made. The complete Console information shown would be this:

```
Give me the next grade in [0 to 10]:6
Give me the next grade in [0 to 10]:6
Give me the next grade in [0 to 10]:7
Give me the next grade in [0 to 10]:8
Give me the next grade in [0 to 10]:10
Sorted grades: [6, 6, 7, 8, 10]
Pick 'a' for mean, 'b' for median, 'c' for mode: a
picked: Mean
Pick '1' for print average, or '2' for plot average: 3
Error in pick_visualization: incorrect option picked
repl process died unexpectedly:
```

Note that the message “repl process died unexpectedly:” is particular to replit when the exit function is used.

Grading

Grading criteria:

General

The submission:

- **IMPORTANT:** If your code does not compile, you lose 50% of your grade so make sure you run your code often (to avoid syntax or runtime errors) and you always have it in a “running” state, even if it does not get the desired results.
- Your grade will be the percentage of tests you pass.

* runs without syntax errors (or -50%) * adds a few small but informative comments (or -10%)

Operations

The program:

- Passes all tests (or lose 5% per missed test).

Submitting

I will collect all of your hw03 directory files at the due date/time so make sure it is complete and running by then.