



uOttawa

ITI 1120 E - Winter 2023 - Assignment 2

The purpose of this assignment is to practice concepts that you have seen in class. Besides float and Boolean, the only collection you can use is strings. You are **not permitted to use any other collection (such as list/array, tuple, or dictionary)** in this assignment. Thus this assignment does not require the use of recursion and global variables. The question that breaks this rule will receive zero since they suggest the required understanding of the material covered so far. Assignment 2 is 5% of the final mark. The deadline for all regular and SASS students is **Sunday, February 12, 2023, at 11:59 pm**. No late assignments will be accepted.

1. This work must be done **individually**.
2. Submit a zip file in BrightSpace **a2\_xxxx.zip** where the xxxx corresponds to your student id. The zip file should contain the files a2\_xxxx.py, and a2\_xxxx.txt.
3. Put all the required functions in a file a2\_xxxx.py and the results of your tests are a file a2\_xxxx.txt.
4. Your grade may partially be determined by automatic (unit) tests that will test your functions.
5. All the specified requirements below are mandatory (including function names, and the behavior implied by examples). Any requirement that is specified and not met will result in a deduction of points. A function that does not have the required name will get zero points.
6. You **MUST** add comments for pre-conditions and assumptions in addition to the doctoring for each function (with a short description, the pre-conditions, and the **type contract**).
7. If your file a2\_xxxx.py gives a syntax error, the mark may be zero.
8. Note that we will be using a plagiarism detection tool. In case two assignment solutions are identical or very similar, the mark will be zero for both students.

Any student caught in plagiarism will receive zero for the whole assignment and will be reported to the dean. Finally, showing/giving any part of your assignment code to a friend also constitute plagiarism and the same penalties will apply. You are not allowed to consult any tutoring website.

## Section1: A Library of Functions

**Question 1. (5 points)** Assume that the Rideau Canal opens for skating once the ice surface is at least 30 centimetres thick and the average temperature in the past 10 days was less or equal to  $-10^{\circ}\text{C}$ . Write a Python function called **skate** that takes two inputs (the thickness of the ice in cm, and the average temperature in the past 10 days in Celsius) and returns True if the canal opens for skating and False otherwise. See examples of using the function at the end of this file.

**Question 2. (5 points)** Write a Python function called **alphaNote** that takes one input (a float between 0 and 100, inclusively) and returns a letter mark. No need to check if the input is between 0 and 100. Assume it is, for this question. You will check this in Question 3.

Use the following (uOttawa) letter grade correspondence:

A	85-89	A-	80-84
B+	75-79	B	70-74
C+	65-69	C	60-64
D+	55-59	D	50-54
E	40-49	F	0-39

**Question 3. (5 points)** Implement a function called **alphaNoteVerif** that uses the function from Question 2, but it reads the keyboard to get a numeric mark, checks if the mark is between 0 and 100 (including 0 and 100). If it is not, it repeatedly asks the user to re-input the mark until it is a value in the expected range. Assume that the user is inputting an integer or real number, no need to check this for now. The function should call the function **alphaNote** from question 2 and print a message with its result. In addition, it should print the message: “Failed” if the letter grade was “E” or “F”, or the message “Passed” if it was any of the other values.

**Question 4. (5 points)** Implement a function called **loops** that takes an integer **n** and displays every second value, from 1 to n, then from n to 1. Take a look at the examples provided in section 2.

**Question 5. (5 points)** Write one function called **test\_password** that takes any user input password and verify if the password meets the following requirements or not: length between 8 and 16 alphanumeric characters, and needs to include at least, one lower case letter, one upper case letter, number, and special characters (only @, -, #, \$, or % are accepted). To verify your work implement a function called **tester** that asks the user to interactively enter the password and verify all the requirements are met or not. Your function should print statements that tell if the password is acceptable or not. Take a look at the examples provided in section 2.

## Section2: Testing

Test all the functions in the Python interpreter and add the results in the file a2.txt

Copy and paste is the file a2.txt when you test each function in the interpreter. Your file a2.txt should look something like this (**test with other values too**):

### #test Q1

```
>>>skate(30, -10)
True
>>>skate(25.4, -15)
False
>>>skate(33, -15)
True
>>>skate(33, -5)
False
```

### #test Q2

```
>>>alphaNote(100)
'A+'
>>>alphaNote(89)
'A'
>>>alphaNote(56)
'D+'
>>>alphaNote(30)
'F'
```

### #test Q3

```
>>>alphaNoteVerif()
Please input the final mark (from 0 to 100): 103
Please input the final mark (from 0 to 100): 200.6
Please input the final mark (from 0 to 100): 78
The letter mark is: B+
Passed
```

```
>>>alphaNoteVerif()
Please input the final mark (from 0 to 100): -8
Please input the final mark (from 0 to 100): 0
The letter mark is: F
Failed
```

### #test Q4

```
loops(13)
1 3 5 7 9 11 13
13 11 9 7 5 3 1
loops(10)
1 3 5 7 9
10 8 6 4 2
```

### #test Q5

```
>>>tester()
Enter your password: uOttawa2022
Try again, your password does not meet all requirements
>>>tester()
Enter your password: uOttawa@2022
Great, your password meets all requirements
```