



COMPUTER SCIENCE 10A (SUMMER TERM, 2022)

INTRODUCTION TO PROBLEM SOLVING IN PYTHON

EXAM 3

DEADLINE: FRIDAY AUG 12TH, 5:00PM

Start here: Please include the following pledge in the header comments of your source code file that you will submit for this quiz along with your name that will serve as your signature.

Ethics Pledge: With your signature you are certifying the information above and you also affirm the following: “I agree to complete this quiz without unauthorized assistance from any person, material, or device.”

Instructions: Please read these instructions carefully before you start.

This is a take home quiz. For the purposes of this quiz you will have to write a Python program that will solve the problems described in the next page. You will have 24 hours to implement your solution, and the solution will have to be submitted on Latte. Please see submission instructions in the end of this document

- You are not allowed to use code from online resources.
- You are not allowed to discuss your solution with anybody else, or exchange/share code with any other student
- Your submission will be tested for plagiarism using the MOSS system.
- Any sign of collaboration will result in a 0, and additional sanctions may be applied as described in the Syllabus of the course under Academic Integrity.

General Instructions (20pts)

All problems will be solved in one python file named `yourfirstname_yourlastnameExam3.py`. For every problem below you will create one function that will solve each problem. So, since we have 4 problems, you will define 4 functions, and the main function in your python file. The only line of code outside a function will be a call to your `main()` function. You are not allowed to use global variables. You can only use Python features we have seen in the course. In the `main()` you will create the input that you will need to pass as parameters to each one of the functions, and call the functions to test your code. For the functions that you will call, you can pass parameters and receive return values if you need to. Lastly, you will print the output of your functions to demonstrate the correct operations of your functions.

You have to include a header comment with your name, today's date, and the pledge. You also have to include proper documentation of your code (eg. Add comments before every function, and for lines of code that need clarification).

Example of header comment:

```
# Name: Iraklis Tsekourakis
# Date: Aug 11, 2022
# COSI 10a, Exam 3
# I agree to complete this quiz without unauthorized assistance
# from any person, material, or device.
```

1. (20 pts)

Write a function called `cap` that takes 2D list and a number as parameters. Your function should replace any numbers stored in the passed in list of lists that are greater than the passed in number with the passed in number. Examples are shown in the table below using the below list. Note that the passed in list might be of any size.

```
data = [[18, 14, 29], [12, 7], [2, 22, 5]]
```

call	List of lists after the call
<code>cap(data, 20)</code>	<code>[[18, 14, 20], [12, 7], [2, 20, 5]]</code>
<code>cap(data, 2)</code>	<code>[[2, 2, 2], [2, 2], [2, 2, 2]]</code>
<code>cap(data, 0)</code>	<code>[[0, 0, 0], [0, 0], [0, 0, 0]]</code>

2. (20 pts)

Write a function called `overlap` that takes a set of integers and a list of integers as parameters and that returns a new set containing values that appear in both structures.

For example, given set and list:

```
set1: {0, 19, 8, 9, 12, 13, 14, 15}
list1: [0, 19, 2, 4, 5, 9, 10, 11]
```

If the following call is made:

```
overlap(set1, list1)
```

the function would return:

```
{0, 19, 9}
```

You are not allowed to construct any structures, besides the set you will return, to solve this problem. You may not alter the passed in list or set. **You may not convert the list to a set.**

3. (20 pts)

Write a function called `split` that takes a set of strings as a parameter and that returns the result of splitting the strings into different sets based on the length of the strings. In particular, your function should return a dictionary whose keys are integers and whose values are sets of strings of that length. For example, if a variable called `words` contains the following set of strings:

```
{to, be, or, not, that, is, the, question}
```

then the call `split(words)` should return a dictionary whose values are sets of strings of equal length and whose keys are the string lengths:

```
{2={be, is, or, to}, 3={not, the}, 4={that}, 8={question}}
```

Notice that strings of length 2 like "be" and "is" appear in a set whose key is 2. If the set had instead stored these strings:

```
{four, score, and, seven, years, ago, our, fathers, brought, forth}
```

Then the function would return this dictionary:

```
{3={ago, and, our}, 4={four}, 5={forth, score, seven, years},  
7={brought, fathers}}
```

Your function should construct the new dictionary and each of the sets contained in the dictionary but should otherwise not construct any new data structures. It should also not modify the set of words passed as a parameter.

4. (20 pts)

Write a function called `odds_to_back` that takes a list of integer values as a parameter and that moves all odd numbers to the back of the list, preserving their relative order. For example, if a variable called `list` stores this sequence of values:

```
[7, 2, 8, 9, 4, 13, 7, 1, 9, 10]
```

then the following call:

```
odds_to_back(list)
```

should leave the list with the following values:

```
[2, 8, 4, 10, 7, 9, 13, 7, 1, 9]
```

Notice that the list begins with the even values in their original order followed by the odd values in their original order. You may not construct any extra data structures to solve this problem. You must solve it by manipulating the list you are passed as a parameter.

Submission and Grading:

All your code should be written in one python file named `yourfirstname_yourlastnameExam3.py`, then zip the file into a zip file for submission. (Please make sure to use exactly this file name, including identical capitalization).

Your program should be submitted via Latte the day it is due. **NO LATE SUBMISSION WILL BE ACCEPTED FOR THIS QUIZ/TAKE HOME EXAM.**

You will be graded on:

- **External Correctness:** The output of your program should match exactly what is expected. Programs that do not compile will not receive points for external correctness.
- **Internal Correctness:** Your source code should follow the stylistic guidelines shown in class. Remember to include the comment header at the beginning of your program and comment your code.