

# Honors Computer Science Python

## Mr. Clausen

### Program 6A, 6B, 6C, 6D

#### PROGRAM 6A Rock Paper Scissors Lizard Spock (40 points)

This program is going to practice functions. Write a program to play the game Rock, Paper, Scissors, Lizard, Spock (RPSLS). The user will be playing against the computer. Ask the user for their choice, generate a random choice for the computer, determine who wins or if there is a tie, keep a running total of who wins each round, and display the results each time the user chooses to play the game. This program can be written using if, elif statements to determine the winner, and to convert from the user's choice to a number and to convert the computer's numeric choice to a corresponding string representing the computer's choice. You could use your own algorithm or the class could ask me to discuss an algorithm using the modulus operator.

- 1) Use a `DocString` at the beginning of the program for your comments.
- 2) Initialize all of the variables that are to be used in this program.
- 3) Use print statements to display your name and period output using a function named "displayMyInfo" just like program 5B.
- 4) Display the menu (as pictured below) and ask the user to enter their choice. Upper and lower case values should work for the menu choice to quit.
- 5) Make sure that you use descriptive identifiers for all of your variables.
- 6) You must use functions throughout this program. You must have a main function that declares all the variables (except variables that may be local to a function), and calls all the other functions. Don't forget that the last line of code should be "main()" so the program will run. Remember to include a docstring in every function that includes the "type contract", a description of the function, and a sample function call where applicable.
- 7) After the user has selected their choice, generate a random integer to represent the computer's choice.
- 8) Determine who won or if there is a tie. Keep a running total of how many times the player wins versus the computer wins.
- 9) Display who chose what, who won (or if there was a tie), and how many times the "player" won and how many times the "computer" won.
- 10) Save your program as LastNameFirstNameP6A.py.

```

Rock Paper Scissors Lizard Spock Main Menu
1. Rock
2. Spock
3. Paper
4. Lizard
5. Scissors
Q. Quit the game

```

## PROGRAM 6B Arithmetic Series (20 points)

Use **recursion only** to generate an Arithmetic Series of the numbers from the user's choice of a starting number to the user's choice of an ending number with running sums of the numbers. The user should also be able to choose what should be added to each previous term to get the next term. We called this the Common Difference in your math class. Use a recursive function named **"sum"** to do the work. **Do not use the code from the textbook on page 212. DO NOT use iterative loops of any kind (you cannot use: "for" or "while" loops anywhere in this program)!** You will need an "if statement" to ensure the recursion has a well-defined stopping state.

Print out the results in a nice table format as illustrated below using format strings and operators (Page 85-89).

Term Number	Term	Sum
1	1	1
2	3	4
3	5	9
4	7	16
5	9	25

- 1) Use a `DocString` at the beginning of the program for your comments.
- 2) You must use functions throughout this program. You must have a **"main"** function that initializes all the variables (except variables that may be local to a function), and calls all the other functions. Make sure that you use descriptive identifiers for all of your variables. Don't forget that the last line of code should be `"main()"` so the program will run. Remember to include a docstring in every function that includes the "type contract", a description of the function, and a sample function call where applicable.
- 3) Use print statements to display your name and period output using a function named `"displayMyInfo"` just like program 5B.
- 4) Create separate functions to get and return the user's starting value, the upper limit, and the common difference.
- 5) Make a `"displayHeader"` function to display the titles `"Term Number"`, `"Term"`, and `"Sum"` using format strings and operators.
- 6) Create the recursive function `"sum"` which increments the term number, calculates the term and the running totals (sum) and prints the information formatted nicely using format strings and

operators.

7) Save your program as LastNameFirstNameP6B.py.

### **PROGRAM 6C Functionally I'm Sort of Searching for Monty Python (35 points)**

Rewrite program 5A using functions to practice the sorts and searches that we cover in every programming language. This program will practice the Python list methods as well as the standard algorithms. We are adding the Binary Search (Iteration and Recursion), Selection Sort, Bubble Sort, Insertion Sort, and Quick Sort.

- 1) Use a `DocString` at the beginning of the program for your comments.
- 2) Initialize all of the variables that are to be used in this program in the main function.
- 3) Use print statements to display your name and period output using a function named "displayMyInfo" just like program 5B.
- 4) Display the menu (as pictured below) and ask the user to enter their choice.
- 5) Create a function, `ControlMenuExecution` which contains the "if, elif, else statements" and calls all of the other functions.
- 6) For this program we will implement menu choices for all of the other choices that we have not previously implemented. There are PowerPoint lessons for all of the sorts and searches on my website. Upper and lower case values should work for the menu choices.
- 7) For both of the Binary Searches, we are determining presence or absence of the "target" number. For the Linear Search, be sure to display the index numbers of every occurrence of the "target" number.
- 8) Make sure that you use descriptive identifiers for all of your variables including names for the lists. Please use the names **originalList** and **listCopy**. The word "list" is a reserved word in Python (menu choice #2).
- 9) In menu choice #1, ask the user how many elements they want in the list. Generate random integers from 1 to this value to fill the list with that many elements.
- 10) In menu choice #2, properly copy the list. You will need to copy the list in the main function, otherwise you will get an empty list.
- 11) For menu choice "S" ask the user for the number they wish to find. Display the index number of every occurrence of this number as well as how many times the number was found in the list.
- 12) Save your program as LastNameFirstNameP6C.py.

## I'm Sort Of Searching For Monty Python Part 2

1. Generate Random Numbers For The Original List
2. Copy The Original List
3. Sequential (Linear) Search
4. Binary Search Iteration (The List Must be Sorted)
5. Binary Search Recursion (The List Must be Sorted)
6. Selection Sort
7. Bubble Sort
8. Insertion Sort
9. Quick Sort
- P. Python's Sort Method
- S. Python's Search and Count Method
- I. Insert an element in the List at any position
- D. Delete an element from the List at any position
0. Display the numbers using a 'loop' (press zero)
- Q. Quit

### **PROGRAM 6D What's The Password? (40 points)**

This program is going to practice functions. Write a program that asks the user how many passwords that they wish to generate. Next, ask the length of the passwords and then ask the user if they want to use 1) UPPER and lower case letters only, 2) UPPER and lower case letters and numerals (numbers), and 3) UPPER and lower case letters and numerals (numbers) and symbols. Implement a choice to save these passwords to a text file as well as displaying them on the screen. This program needs to use functions.

- 1) Use a DocString at the beginning of the program for your comments.
- 2) Import random.
- 3) Initialize all of the variables that are to be used in this program in the main function. The use of global variables will result in a grade of "F" for a grade on this program. The "main" function should be the first function listed in your source code. You must have a main function that declares all the variables (except variables that may be local to a function), and calls all the other functions. Don't forget that the last line of code should be "main()" so the program will run.
- 4) Use print statements to display your name and period output using a function named "displayMyInfo" just like program 5B.
- 5) Display the menu (as pictured below) and ask the user to enter their choice. Upper and lower case values for the letter "Q" should work for the menu choice to quit.
- 6) Make sure that you use descriptive identifiers for all of your variables.
- 7) You must use functions throughout this program. Remember to include a docstring in every function that includes the "type contract", a description of the function, and a sample function call where applicable.

8) Save your program as LastNameFirstNameP6D.py.

```
Password Generator Main Menu
```

```
1. Upper and Lower Case Letters
```

```
2. Upper and Lower Case Letters with numbers
```

```
3. Upper and Lower Case Letters with numbers and symbols
```

```
4. Save passwords to a text file
```

```
Q. Quit
```