UNIVERSITY OF CALIFORNIA, MERCED

BERKELEY · DAVIS · IRVINE · LOS ANGELES · MERCED · RIVERSIDE · SAN DIEGO · SAN FRANCISCO

SANTA BARBARA · SANTA CRUZ

CSE22: Introduction to Programming

Final Examination

Fall 2021

Instructions

- This is a closed book exam. No notes and/or electronic devices may be used.
- Answer every question on an answer paper/booklet with your name on it.
- Make sure you number every question accurately on your answer paper.
- You have 3 hour to complete this exam.
- If you are unsure of anything, please ask.

Section	**Points**	**Score**
Linux Command Line	20	
Basic Programming	50	
Problem Solving	30	
Searching and Sorting Algorithms	50	
Complexity Analysis	30	
Recursion	20	
TOTAL	**200**	

1 Linux Terminal Commands [20 points]

1.1. What is the terminal command for navigating to the parent of the folder we are in? [2 points]

1.2. Suppose there is a folder called `stuff` inside your home folder. Assuming we are in some random folder on our hard drive, write the terminal command/s that would display the contents of the `stuff` folder referred to in this question? [3 points]

1.3. Write the terminal commands for performing the following sequence of tasks: [15 points]

- Ensure that we are currently in the home directory
- In the home folder, create two folders, `left` and `right`
- Create a file called `one.txt` in the `left` folder
- Create a file called `two.txt` in the `right` folder
- In the `left` folder, rename the file `one.txt` to `file1.txt`
- Delete the entire `right` folder

2 Basic Programming [50 points]

2.1. Write a Python program that outputs the string `"Hello World!"`. Assuming your program is saved in a file called `hello.py`, and your terminal is currently in the same folder as this program, write the terminal command that you would use to run your program. [5 points]

2.2. Write a Python program that contains the following: [10 points]

- variable called `n` storing the value `17`
- variable called `rate` storing the value `2.3`
- variable called `name` storing the value `"Avery"`
- variable called `found` storing the value `True`
- variable called `lost` storing the value `"False"`

For each one of the variables above, specify their type as a comment next to the definition.

2.3. Write a Python program that asks the user for two numbers. Store them in two variables called `x` and `y`. Then ask the user to input an arithmetical operation, namely `+`, `-`, `*`, `/`, which represent addition, subtraction, multiplication, and division, respectively. Store the operation in a variable called `op`. Finally, output the result of applying the given operation on the two numbers provided. [10 points]

2.4. Write a Python program that asks the user to enter their name. Store their input in a variable called `name`. Calculate the number of vowels in their name. Store that result in a variable called `vowelCount`. As a reminder, the vowels are "A", "E", "I", "O", and "U". Output a message for the user, of the form `Your name contains <x> vowel/s`, where `<x>` is the number of vowels in their name. [10 points]

2.5. Write a Python program that asks the user to enter some numbers and then adds them up, printing our the result. The user should be allowed to enter as many numbers as they wish, and can indicate that they are done by entering -1. The -1 should not be used in the calculation.

Your program should store these numbers in a list called `nums`. As the user enters the numbers, your program should be adding them to the list.

Once the numbers have all been entered, your program should add them up, storing the result in a variable called `total`.

Finally, your program should output a message of the form `Your numbers add up to <x>`, where `<x>` is the sum of all elements in the `nums` list. [15 points].

3 Problem Solving [30 points]

3.1. Write a Python function that takes in a list of numbers and returns the average. [10 points]

3.2. Write a Python function that takes a list of numbers that are sorted in order from smallest to largest and removes the duplicate elements. The function should return the resulting list. [10 points]

3.3. Write a Python function that takes in an English sentence, stored in a string `s`, and a number `k`. The function should return the number of words that are strictly longer than `k`. [10 points]

4 Complexity Analysis [30 points]

4.1. Using comparisons as the operation of interest, determine the complexity function of the code snippet below. Include both the best and the worst case performance, and indicate the conditions under which the best and worst cases manifest themselves. Provide justification for your answer. [10 points]

```
1:  def doSomething(someList):
2:      x = someList[0]
3:      for i in range(1, len(someList)):
4:          if someList[i] == x:
5:              return True
6:      return False
```

4.2. Using comparisons as the operation of interest, determine the complexity function of the code snippet below. Provide justification for your answer [5 points]

```
1:  def someFunction(someList):
2:      for i in range(len(someList)):
3:          for j in range(1,len(someList)):
4:              if someList[j-1] > someList[j]:
5:                  temp = someList[j]
6:                  someList[j] = someList[j-1]
7:                  someList[j-1] = temp
8:      return someList
```

4.3. Using comparisons as the operation of interest, determine the complexity function of the code snippet below. Include both the best and the worst case performance, and indicate the conditions under which the best and worst cases manifest themselves. Provide justification for your answer. [10 points]

```
1:  def doSomething(someList):
2:      for i in range(len(someList)):
3:          if someList[i] == x:
4:              return True
5:          else:
6:              return False
```

4.4. Prove that the complexity function $f(n) = 2n^2 + 5n + 2 \in O(n^2)$. [5 points]

5 Searching and Sorting Algorithms [50 points]

5.1. Write a Python function that implements the linear search algorithm. It should accept a list of numbers and a key and should return the position in the list where key is found. If key is not found, the function should return -1. [5 points]

5.2. Provide a list of 5 integers that will cause linear to perform in the its best case. [5 points]

5.3. Provide a list of 5 integers that will cause linear to perform in the its worst case. [5 points]

5.4. Write a Python function that implements selection sort. It should accept a list of numbers and should return the same list but in sorted. [10 points]

5.5. Write a Python function that implements insertion sort. It should accept a list of numbers and should return the same list but in sorted. [10 points]

5.6. Provide a list of 5 integers that will cause insertion sort to perform in the its best case. [5 points]

5.7. Provide a list of 5 integers that will cause insertion sort to perform in the its worst case. [5 points]

5.8. Write a Python function that implements binary search. It should accept a list of numbers and a key, and should return True if the key appears in the list, and False otherwise. [5 points]

6 Recursion [20 points]

6.1. Given that $f(n) = 2f(n-1) + 1$ and $f(1) = 1$, calculate $f(5)$. Show all your work. [10 points]

6.2. Write a recursive Python function that takes in a string and returns `True` if the string is a palindrome, and `False` otherwise. [10 points]