Programming for Bioinformatics | BIOL7200

Week 7 Exercise

October 5, 2021

Instructions for submission

Submit individual scripts for each exercise on Canvas. Name each script file as <gtusername>_1.py, <gtusername>_2.py, <gtusername>_3.py, and so on. Please make sure you include your GT username while naming your scripts. We realize there are a lot of exercises, but they aren't too long.

Grading Rubric

The assignment will be graded out of 100.

- Questions 1 -13 carry 3 points each (39)

- Questions 14-21 carry 5 points each (40)

- Questions 22-24 carry7 points each (21)

Exercises

There are several ways of doing the following series of problems. Adopt the simplest one. Most of these questions' solutions are within 10-20 lines.

1. Using a while loop, print whether a number is even or odd while counting from 0 to 49 by 3.
2. Using a loop, print the first 10 numbers of the Fibonacci sequence (1, 1, 2, 3, 5, 8, 13, 21, 34, 55, …)
3. Use a while loop to print the factorial of a number
4. Find all the numbers between 2000 and 3000 that are divisible by 7 but not by 5
5. Using a while loop and input(), check if an entered string matches "secretpassword". Once the match is found, exit the loop.
6. Write a loop that generates a triangle of text, *i.e.*

```
*
**
***
****
*****
****
***
```

```
**
*
```

7. Swapping values:
    a. Create two numeric variables: **a = 10** and **b = 30**
    b. Swap their values by using a temporary variable
    c. Swap their values without using any intermediate variable
8. Use **input** to take a series of numbers from the user (one number at a time); stop if the user enters **0** and return the sum, average, maximum and minimum of the numbers received.

Logic problems

9. Given two numeric values (say **a** and **b**), return their sum unless they are equal (i.e., **a+b**). If they are equal, return their sum plus the square of the two numbers (i.e., **a+b+a*a+b*b**).
10. The police in your city aggressively ticket speeders, unless it happens to be their birthday. There is an unofficial policy to subtract 5mph from your speed, which can be the difference between a ticket and no ticket, on your birthday. Fines for speeding are small if within 1-15 mph over and large if > 15mph over. Return the expected fine ("none", "small", "large") depending on your speed and whether it is your birthday.

String manipulation

11. Given two strings, one of length 4 (**string1**) and one of any length (**string2**), return **string2** in the middle of **string1**.
    a. string1 = "****"
    b. string2 = "Bold move, Cotton"
    c. output: **Bold move, Cotton**
12. Given a string of any length, return the first half of the string (For strings of odd length, return the "half + 1" part of the string)
13. Given a string and a logical value, return the reversed string if **TRUE** and return the string as is if **FALSE**.

List manipulation

14. Give an integer, **n**, return a list containing the first **n** numbers of the Fibonacci sequence.
15. Given two lists, return **TRUE** if the sums of each list are equal, else return **FALSE**.
16. Given a list of integers and two numeric inputs (**A** and **B**), return the sum and product of the integers at positions **A** and **B** in the list.
17. Given two lists of numbers, return a list containing the median number from each list (Assume the lists will always be an odd length)

Dictionary manipulation

18. Given an integer, **n**, write a program that generates a dictionary that contains as keys the **1:n** and values equal to **n*n**. For example:
    Input: **4**
    Output: **{1:1, 2:4, 3:9, 4:16}**

Multiple dimensions
Create a 3 x 5 matrix of numbers (of your choice) to use for these exercises.

19. Write a program that computes the mean of each column and the mean of means
20. Write a program that computes that pairwise difference and returns the average difference. For example: if position **(1,1)=10** and **(1,2)=7**, the pairwise distance is **3**.
21. Write a program that reverses the order of numbers in each row

Advanced string and list problems

22. Given a string made up of **{** and **}**, write a script that determines if all the opened curly braces have a paired closing brace. Ensure the pairing component, **{}{}** have all curly braces in pairs whereas **}}{{{}** do not have them in pairs. A simple counting of opening and closing curly braces will return pairing for both cases which will be incorrect! Use the following test cases to see if your code works properly:

Case	Input	Paired?	Case	Input	Paired?
(a)	{{{}}}	Yes	(d)	}}{{	No
(b)	{}{}}	Yes	(e)	{}{	No
(c)	{{{}}{}}	Yes	(f)	{{}}{	No

23. Detect if an input string is an English palindrome or not. *I.e.,* words or sequence that reads the same backwards as forwards. E.g., madam or Anna
24. Detect if an input string is a biological palindrome or not. Biological palindromes are self-complimentary sequences. *E.g.,* **AACAGTTTATAAACTGTT** (**AACAGTTTA** and its reverse compliment **TAAACTGTT**) or **ACACTGT**