

Homework 5 Report -- Group 334-7

HW2	Driver - Cameron Rabiyan	Navigator - Maya Apotheker	Navigator - Manny Gamboa
Coding	66.6%	0.0%	33.3%
Results	33.3%	33.3%	33.3%
Report	0%	50.0%	50.0%

Introduction: In homework 5, we were tasked with building many different programs that test our skills using Python for problems 1-7. For problem 8, we were tasked with using python and numpy to solve a factorial and to solve the taylor series of different functions using a recursive implementation.

Problem 1: For problem 1, we printed out the phrase “yourFirstName” with enough leading spaces to place that letter in the 70th column. We did this using the format the position using ‘%70s’.

Problem 2: For problem 2, we were tasked with using the do_twice function to print the phrase ‘spam’ multiple times. We then had to rearrange a preprogrammed file titled do_twice so that we can call print_twice two times, resulting in 4 prints of the phrase ‘spam.’ First, we modify our do_twice file that calls our function twice, taking our parameter s, defined as ‘spam,’ twice. Then we wrote print_spam that takes strings as parameters and prints them twice. Then we use do_twice to call print_twice two times, passing ‘spam’ as an argument to produce 4 total prints of ‘spam’

Problem 3: In problem 3, we wrote a program that takes a list and parameters and checks whether it is in ascending order. We defined our string l as a parameter defined [1,2,3]. Using if, elif, else logic, we created a program that confirms that the string [1,2,3] and confirms that it is in ascending order.

Problem 4: In problem 4, we were tasked with finding the cumulative sum of a list of numbers predefined as [1,2,3]. This should result in [1,3,6]. First, we defined our string as a parameter, defined as [1,2,3]. Then, we made a shallow copy of our list to pass into a function. Then, we created a function that passes the values of our string through a program that sums the respective components.

Problem 5: In problem 5, we wrote a program that takes the imputed first and last name as a list and capitalizes the first character of the first and last name of someone in our group. In our case, cameron rabiyan. First, we create a function that capitalizes the first letter of the first and last name defined in our string. Then, we implement a function that rearranges the name as Rabiyan, Cameron if a False argument is passed alongside the list. Then, we implement a function that prints out the full name, where both the

Problem 6: In problem 6, we wrote a program that confirms whether a string is a palindrome or not. A palindrome is a word that is spelled the same backward as it is forward. First, we define the input as a variable that can be passed. Then, we passed this variable into our function verify_palindrome, which verifies the existence of a palindrome in the string and prints True or False accordingly.

Problem 7: In problem 7, we wrote a program that verifies that 3 different points on the x-y plane and checks if these 3 points can form a triangle. First, we predefine the maximum amount of inputs within total_numbers as 3, as a triangle requires 3 points on an x-y plane. Then, we pass total_numbers into get_realnumber which collects 3 sets of x-y points from the user. Finally, we pass get_realnumber into our function to is_triangle which verifies whether these user-inputed points can form a triangle on the x-y plane.

Problem 8(Taylor): In part 1 of problem 8, we wrote a program that implements a function exp_appx(x,n) that returns an approximation of exp(x) based on the summation of the taylor series to the n-th degree as part of the much larger infinite summation. First, we wrote a function exp_appx(x,n), which passes the numerator and factorial (which is also the denominator) Then, we added factorial.py from our lectures notes into the factorial file directory, this will then be imported into our taylor_serier.py file which allows for access to the math modules needed. This code can then recursively calculate an approximation of the exp_appx(x,n).