Coding Tasks for Practice Homework

This handout contains a number of coding tasks. Try to solve each problem in Python. Give the file the heading for each task. Upload the files to the homework assignments when you complete them.

## sub.py

Write a program that asks the user for two numbers, subtracts the smaller number from the larger, and displays the difference.

## lowest\_num.py

Write a program that defines a function to compare two numbers entered by the user and return the lower one. Display the result.

## lowest\_num3.py

Write a program that asks the user to enter two numbers, defines a function to return the lowest, and then displays the result.

## highest\_num.py

Write a program that asks the user to enter three numbers and outputs the highest number entered.

## highest2.py

Write a program that asks the user to enter three numbers, uses a function to determine the highest, and then prints the result.

## odd\_number.py

Write a program that repeatedly asks the user to enter a number and checks if it is odd or even. The program should stop when the user types 'STOP'.

## odd\_number2.py

Write a program that asks the user to repeatedly enter a number and checks if it is odd or even. The program should run continuously until manually stopped.

## for\_loop\_basic\_numbers.py

Write a program that asks the user for a start and stop value, then uses a for loop to print all the numbers from the start value up to the stop value (inclusive).

## total\_array.py

Write a program that stores an array of numbers and calculates the total sum of all the elements in the array.

## lowest\_in\_array.py

Write a program that stores an array of numbers and finds the lowest number. Then, create a function that repeats this process for an array passed in as an argument.

## binary\_to\_denary.py

Write a program that asks the user for a 4-bit binary string (e.g., '1010') and converts it into its denary (decimal) equivalent.

## denary\_to\_binary.py

Write a program that asks the user for a denary number between 0 and 15 and then outputs its 4-bit binary equivalent.

## 2bithexToDenary.py

Write a program that takes a two-digit hexadecimal value (using digits 0-9 and letters A-F), converts it into a denary (decimal) number, and displays the result.

## circle\_area\_function.py

Write a program that defines a function to calculate the area of a circle given its radius. Ask the user for the radius, call the function, and display the result.

## number\_guesser.py

Write a program that generates a random number between 1 and 10. The user must keep guessing until they enter the correct number.

## unlimited\_logins.py

Write a program that repeatedly asks the user to enter a PIN until they enter the correct one (e.g., '1984'). When the correct PIN is entered, display 'Logged in' and stop the program.

## unlimited\_logins1.py

Write a program that repeatedly asks the user to enter a PIN until it matches the correct one (e.g., '1984'). If incorrect, display 'Incorrect'. When correct, display 'Logged in'.

## login\_basic2.py

Write a program that asks the user to enter a username and password. If the credentials match the stored ones, display 'Logged in'. Allow up to three attempts before the program ends.

## username\_generator.py

Write a program that generates a username from the user’s first name, surname, and year they joined the school. The username should be made up of the last two digits of the year, the first letter of the first name, and the full surname.

## discount\_2dp.py

Write a program that asks the user for an order total and whether they have a discount code. If a valid discount code is entered (e.g., 'summer10', 'welcome', 'refer20'), apply the discount and display the new total rounded to 2 decimal places.

## linear\_search\_name\_finder.py

Write a program that stores a list of names and uses a linear search to find a target name entered by the user. Display the position of the name if it is found, or a message if it is not.

## linear\_2D.py

Write a program that stores exam results for a group of students in a two-dimensional list. The program should ask the user to enter a student's name and an exam number, then output that student's score for the chosen exam. If the name is not found, print an error message.

## linear\_2D\_challenge\_solved.py

Write a program that stores exam results for a group of students in a two-dimensional list and defines a function to find a specific student's result for a given exam. The program should ask the user for a name and exam number, then call the function to display the result.

## 2D\_total.py

Write a program that stores exam results for several students in a two-dimensional list. For each student, calculate the total of their exam scores and display it.

## shopping.py

Write a program that repeatedly asks the user to enter shopping list items until they type 'End'. Store the items in a list and then write the list to a file called shopping\_list.txt.

## shopping adding.py

Write a program that reads a shopping list from a file (see below), allows the user to add more items until they type 'End', and then writes the updated list back to the file.

The file should contain the following text:

['Peas', 'Mushrooms', 'Pork', 'Rice', 'Oil', 'Tomatoes']