Lab Exercises - 2

Python Repetition

For Loops Exercises

1 - Write a programme that asks the user to enter their name and a number. If the number is less than 10, then display their name that number times; otherwise, display the message “Too high” three times.

```
name = input("Enter your name: ")
num = int(input("Enter a number: "))
if num < 10:
    for i in range (0, num):
        print(name)
else:
    for i in range (0,3):
        print("Too high")
```

2 – Set a variable called total to 0. Your programme should ask the user to enter five numbers and after each input, ask them if they want that number to be included. If they do, the programme should then add the number to the total. If they do not want it included, the programme shouldn’t add the number to the total. After the user has entered all five numbers, display the total.

```
total = 0
for i in range (0,5):
    num = int(input("Enter a number: "))
    ans = input("Do you want this number included? (y/n) ")
    if ans == "y":
        total = total + num
print(total)
```

3 - Write a programme that asks which direction the user wants to count (up or down). If they select up, then the programme should ask them for the top number and count from 1 to that number. If the user selects down, the programme should ask them to enter a number below 20 and then count down from 20 to that number. If the user entered something other than up or down, display the message “I don’t understand”.

```
direction = input("Do you want to count up or down? (u/d) ")
if direction == "u":
    num = int(input("What is the top number? "))
    for i in range(1, num+1):
        print(i)
elif direction == "d":
    num = int(input("Enter a number below 20: "))
    for i in range(20,num-1, -1):
        print(i)
else:
    print("I don't understand")
```

While Loop exercises

4 - Write a programme that asks the user to enter a number and then enter another number. The programme should add these two numbers together and then ask if they want to add another number. If the user enters “y”, the programme will ask them to enter another number and will keep adding numbers until the user stops answering “y”. Once the loop has stopped, display the total.

```
num1 = int(input("Enter a number: "))
total = num1
again = "y"
while again == "y":
    num2 = int(input("Enter another number: "))
    total = total + num2
    again = input("Do you want to add another number? (y/n) ")
print("The total is ", total)
```

5 – Create a variable called compnum and set the value to 50. Ask the user to enter a number. While their guess is not the same as the compnum value, tell them if their guess is too high or too low and ask them to have another guess. If they enter the same value as compnum, display the message “Well done, you took [count] attempts”.

```
compnum = 50
guess = int(input("Can you guess the number I am thinking of? "))
count = 1
while guess != compnum:
    if guess < compnum:
        print("Too low")
    else:
        print("Too high")
    count = count+1
    guess = int(input("Have another guess: "))
print("Well done, you took ", count, "attempts")
```

6 – Using the song “10 green bottles”, display the lines “There are [num] green bottles hanging on the wall, [num] green bottle hanging on the wall, and is 1 green bottle should accidentally fall”. Then ask the question “how many green bottles will be hanging on the wall?” If the user answers correctly, display the message “There will be [num] green bottles hanging on the wall”. If they answer incorrectly, display the message “No, try again” until they get it right. When the number of green bottles gets down to 0, display the message “There are no more green bottles hanging on the wall”.

```
num = 10
while num >0:
    print("There are ", num, "green bottles hanging on the wall.")
    print(num, "green bottles hanging on the wall.")
    print("And if 1 green bottle should accidentally fall, ")
    num = num - 1
    answer = int(input("How many green bottles will be hanging on the wall? "))
    if answer == num:
        print("There will be ", num, "green bottles hanging on the wall.")
    else:
        while answer != num:
            answer = int(input("No, try again: "))
print("There are no more green bottles hanging on the wall.")
```

Function Exercises

7 – Define a function that will ask the user to enter a number and save it as the variable “num”. Define another function that will use “num” and count 1 to that number.

```
def ask_value():
    num = int(input("Enter a number: "))
    return num

def count(num):
    n = 1
    while n <= num:
        print(n)
        n = n + 1

def main():
    num = ask_value()
    count(num)

main()
```

8 – Create a programme that will allow the user to easily manage a list of names. You should display a menu that will allow them to add a name to the list, change the name in the list, delete a name from the list from the list or view all the names in the list. There should be a menu option to allow the user to end the programme. If they select an option that is not relevant, then it should display a suitable message. After they have made a selection to either, add a name, change a name delete a name or view all the names, they should see the menu again without having to restart the programme. The programme should be as easy to use as possible.

```
def add_name():
    name = input("Enter a new name: ")
    names.append(name)
    return names

def change_name():
    num = 0
    for x in names:
        print(num, x)
        num = num +1
    select_num = int(input("Enter the number of the name you want to change: "))
    name = input("Enter new name: ")
    names[select_num] = name
    return names

def delete_name():
    num = 0
    for x in names:
        print(num, x)
        num = num + 1
    select_num = int(input("Enter the number of the name you want to delete: "))
    del names[select_num]
    return names

def view_names():
    for x in names:
        print(x)
    print()

def main():
    again = "y"
    while again == "y":
        print("1) Add a name")
        print("2) Change a name")
        print("3) Delete a name")
        print("4) View a name")
        print("5) Quit")
        selection = int(input("What do you want to do? "))
        if selection == 1:
            names = add_name()
        elif selection == 2:
            names = change_name()
        elif selection == 3:
            names = delete_name()
        elif selection == 4:
            names = view_names()
        elif selection == 5:
            again = "n"
        else:
            print("Incorrect option: ")
        data = (names, again)
names = []
main()
```

Random Exercises

9 – Write a programme that displays a random fruit from a list of 5 fruits.

```
import random
### Random exercises
fruit = random.choice(['apple', 'orange','mango','grape','banana'])
print(fruit)
```

10 – Write a basic programme that randomly chooses either heads or tails (“h” or “t”). The programme asks the user to make a choice. If their choice is the same as the randomly selected value, display the message “You win”, otherwise display “Bad luck”. At the end, tell the user if the computer selected heads or tails.

```
coin = random.choice(["h", "t"])
guess = input("enter (h)ead or (t)ail: ")
if (guess == coin):
    print("you win")
else:
    print("bad luck")
if coin == "h":
    print(" the computer selected head")
else:
    print("The computer selected tail")
```

11 - Write a programme that randomly chooses a number between 1 and 5. The programme should ask the user to pick a number. If they guess correctly, display the message “Well done”, otherwise the programme should tell them if they are too high or too low and should ask them to pick a second number. If they guess correctly on their second guess, display the message “Correct”, otherwise display “You lose”.

```
num = random.randint(1,5)
guess = int(input("Enter a number: "))
if guess == num:
    print("Well done")
elif guess > num:
    print("Too high")
    guess = int(input("Guess again: "))
    if guess == num:
        print("Correct")
    else:
        print("You lose")
elif guess < num:
    print("Too low")
    guess = int(input("Guess again: "))
    if guess == num:
        print("Correct")
    else:
        print("You lose")
```

12 – Write a programme that displays five colours and asks the user to pick one. If they pick the same as the programme has chosen, say “Well done”, otherwise display a witty answer which involves the correct colour, e.g. “I bet you are GREEN with envy” or “You are probably feeling BLUE right now”. The programme should ask them to guess again; if the user has still not got it right, it keeps giving them the same clue and it asks the user to enter a colour until they guess it correctly.

```
colour = random.choice(["red", "blue", "green", "white", "pink"])
print("Select from red, blue, green, white or pink")
tryagain = True
while tryagain == True:
    theirchoice = input("Enter a colour: ")
    theirchoice = theirchoice.lower()
    if colour == theirchoice:
        print("Well done!")
        tryagain = False
    else:
        if colour == "red":
            print("I bet you are seeing RED right now")
        elif colour == "blue":
            print("Don't feel BLUE")
        elif colour == "green":
            print("I bet you are GREEN with envy right now.")
        elif colour == "white":
            print(" Your brain is blank as a WHITE sheet, as you did not guess correctly" )
        elif colour == "pink":
            print("Shame you are not feeling in the PINK, as you got it wrong")
```

Turtle Graphics exercises

13 - Draw a circle.

```
import turtle

for i in range (0,360):

    turtle.forward(1)
    turtle.right(1)

turtle.exitonclick()
```

14 - Draw an octagon that uses a different colour (randomly selected from a list of six possible colours) for each line.

```
import turtle
import random

turtle.pensize(3)

for i in range (0,8):
    turtle.color(random.choice(["red","blue","yellow","green","pink","orange"]))
    turtle.forward(50)
    turtle.right(45)
```

15 - Draw a pattern that will change each time the programme is run. Use random function to pick the number of lines, the length of each line and the angle of each turn.

```
import turtle
import random

lines = random.randint(5,20)

for x in range (0,lines):
    length = random.randint(25,100)
    rotate = random.randint(1,365)
    turtle.forward(length)
    turtle.right(rotate)

turtle.exitonclick()
```

Data structure exercises

Data structure – List Exercise

16 - Create a list of four three-digit numbers. Display the list to the user, showing each item from the list on a separate line. Ask the user to enter a three-digit number. If the number they have typed in matches one in the list, display the position of that number in the list otherwise display the message "That's not in the list"

```
nums = [123,345,234,765]
for i in nums:
    print(i)
selection = int(input("Enter a number from the list: "))
if selection in nums:
    print(selection, "is in position", nums.index(selection))
else:
    print("That is not in the list")
```

17 - Write a programme that asks the user to enter the names of three people they want to invite to a party and store them in a list. After they have entered all three names, ask them if they want to add another. If they do, allow them to add more names until they say "no". When they say "no", display how many people they have invited to the party.

```
name1 = input("Enter a name of somebody you want to invite to your party: ")
name2 = input("Enter another name: ")
name3 = input("Enter a third name: ")
party = [name1,name2,name3]
another = input("Do you want to invite another (y/n)?: ")
while another == "y":
    newname = party.append(input("Enter another name: "))
    another = input("Do you want to invite another (y/n)?: ")
print("You have ", len(party), "people coming to your party")
```

Data structure – Dictionary Exercises

18 - Create a programme that asks the user to enter four of their favourite foods and store them in a dictionary so that they are associated with keys starting from 1. Display the dictionary in full, showing the key-value pairs. The programme should ask the user which key-value pair they want to get rid of and remove it from the dictionary. Finally, the programme should sort the remaining data and display the dictionary.

```
food_dictionary = {}
food1 = input("Enter a food you like: ")
food_dictionary[1] = food1
food2 = input("Enter another food you like: ")
food_dictionary[2] = food2
food3 = input("Enter another food you like: ")
food_dictionary[3] = food3
food4 = input("Enter another food you like: ")
food_dictionary[4] = food4
print(food_dictionary)
dislike = int(input("Which of these do you want to get rid of? "))
del food_dictionary[dislike]
print(sorted(food_dictionary.values()))
```

Data structure – tuple exercises

19 - Create a tuple containing the names of five countries and display the whole tuple. Ask the user to enter one of the countries that have been shown to them and then display the index number (i.e. position in the list) of that item in the tuple.

Next, the programme should ask the user to enter a number (index) and display the country in that position.

```
country_tuple = ("Cameroon", "Equatorial Guinea", "Guinea-Bissao", "Papua New Guinea", "Guinea-Conakry")
print(country_tuple)
print()
country = input("Please enter one of the countries from above: ")
print(country, "has index number", country_tuple.index(country))
print()
num = int(input("Enter a number between 0 and 4: "))
print(country_tuple[num])
```

Numeric Arrays

20 – Write a programme that asks the user for a list of five integers. Store them in an array. Sort the list and displays it in reverse order.

```
from array import *

nums = array('i',[])

for i in range (0,5):
    num = int(input("Enter a number: "))
    nums.append(num)
nums = sorted(nums)
nums.reverse()

print(nums)
```

21 – Create an array which will store a list of integers. Generate five random numbers and store them in the array. Display the array (showing each item on a separate line).

```
from array import *
import random

nums = array('i',[])

for i in range (0,5):
    num = random.randint(1,100)
    nums.append(num)

for i in nums:
    print(i)
```

22 – Create two arrays (one containing three numbers that the user enters and one containing a set of five random numbers). Join these two arrays together into one large array. Sort this large array and display it so that each number appears on a separate line.

```
from array import *
import random

num1 = array('i',[])
num2 = array('i',[])


for i in range (0,3):
    num = int(input("Enter a number: "))
    num1.append(num)

for i in range (0,5):
    num = random.randint(1,100)
    num2.append(num)

num1.extend(num2)
num1 = sorted(num1)

for i in num1:
    print(i)
```

23 – Ask the user to enter five numbers. Sort them in order and present them to the user. Ask them to select one of the numbers. Remove it from the original array and save it in a new array.

```
from array import *
import random

nums = array('i',[])


for i in range (0,5):
    num = int(input("Enter a number: "))
    nums.append(num)
nums = sorted(nums)
for i in nums:
    print(i)

num = int(input("Select a number from the array: "))
if num in nums:
    nums.remove(num)
    num2 = array('i',[])
    num2.append(num)
    print(nums)
    print(num2)
else:
    print("That is not a value in the array")
```

24 – Display and array of five numbers. Ask the user to select one of the numbers. Once they have selected a number, display the position of that number in the array. It they enter something that is not in the array, ask them to try again until they select a relevant number.

```
nums = array('i',[4,6,8,2,5])


for i in nums:
    print(i)

num = int(input("Select one of the numbers : "))
tryagain = True
while tryagain == True:
    if num in nums:
        print("This is in position", nums.index(num))
        tryagain = False
    else:
        print("Not in array")
        num = int(input("Select on of the numbers"))
```

25 – Create an array of five numbers between 10 and 100 which each have two decimal places. Ask the user to enter a whole number between 2 and 5. If they enter something outside of that range, display a suitable error message and ask them again to try until they enter a valid number. Divide each of the numbers in the array by the number entered and display the answers shown to two decimal places.

```
from array import *
import math

nums = array('f',[34.76,35.56,67.09, 23.12, 89.34])
tryagain = True
while tryagain == True:
    num = int(input("Enter a number between 2 and 5: "))
    if num <2 or num >5:
        print("Incorrect value, try again.")
    else:
        tryagain = False
for i in range(0,5):
    ans = nums[i]/num
    print(round(ans,2))
```

2 D Lists and Dictionaries

26 – Create the following using a simple 2D list and ask the user to select a row and a column and display that value.

	0	1	2
0	2	5	8
1	3	7	4
2	1	6	9
3	4	2	0

```
list = [[2,5,8],[3,7,4],[1,6,9],[4,2,0]]
row = int(input("Select a row: "))
col = int(input("Select a column: "))
print(list[row][col])
```

27 - Using the 2D list in the previous exercise, ask the user which row they would like displayed and display just that row. Ask them to enter a new value and add it to the end of the row and display row again.

```
list = [[2,5,8],[3,7,4],[1,6,9],[4,2,0]]
row = int(input("Select a row: "))
print(list[row])
newvalue = int(input("Enter a new number: "))
list[row].append(newvalue)
print(list[row])
```

28 – Change the previous programme to ask the user which row they want displayed. Display that row. Ask the user which column in that row they want displayed and display the value in held in there. Ask the user if they want to change the value and change the data. Finally, display the whole row again.

```
list = [[2,5,8],[3,7,4],[1,6,9],[4,2,0]]
row = int(input("Select a row: "))
print(list[row])
col = int(input("Select a column: "))
print(list[row][col])
chnage = input("Do you want to chnage the value? (y/n) ")
if chnage == "y":
    newvalue = int(input("Enter new value: "))
    list[row][col] = newvalue
print(list[row])
```

29 – Ask the user to enter the name age and show size for the four people in a list and display the names and ages of all the people in the list but do not show their show size.

```
list = {}
for i in range(0,4):
    name = input("Enter name: ")
    age = int(input("Enter age: "))
    shoe = int(input("Enter showe size: "))
    list[name] = {"Age":age,"Shoe size":shoe}

for name in list:
    print((name),list[name]["Age"])
```

30 – After gathering the four names, ages and shoe sizes, ask the user to enter the name of the person they want to remove from the list. Delete this row from the data and display the other rows on separate lines.

```
list = {}
for i in range(0,4):
    name = input("Enter name: ")
    age = int(input("Enter age: "))
    shoe = int(input("Enter showe size: "))
    list[name] = {"Age":age,"Shoe size":shoe}

getrid = input("Who do you want to remove from the list? ")
del list[getrid]

for name in list:
    print((name),list[name]["Age"], list[name]["Shoe size"])
```