```
#-------------------------------------------------------------------------
# Name:        Higher 2020 Assignment Program
# Purpose:     Working Example (parallel arrays)
#
# Author:      Mr Simpson
#-------------------------------------------------------------------------

# Additional function to validate password. Called from getMember()
def validatePassword():
    valid = False # Used to continue loop until password is valid
    while valid == False:
        newPassword = str(input("Please enter valid password: "))

        # Test password for first capital and last characters
        if ord(newPassword[0:1]) >= 65 and ord(newPassword[0:1]) <= 90:
            if ord(newPassword[-1:]) == 35 or ord(newPassword[-1:]) == 36 or ord(newPassword[-1:]) == 37:
                # Boolean variable only changes when conditions of both ifs are met
                valid = True
            else:
                # Not required by the task but good practise to feedback to user.
                print("The last character should be #, $ or %")
        else:
            print("The first character should be a capital letter")
    return newPassword

# 1. Get new member first name, surname, category and valid password
def getMember():
    # User inputs the details of the new member
    newFirstName = str(input("Please enter the members first name: "))
    newSurname = str(input("Please enter the members surname: "))
    newCat = str(input("Please enter the members category: "))

    #Call function to validate password
    newPass = validatePassword()

    return newFirstName,newSurname,newCat,newPass

#-------------------------------------------------------------------------
```

Trinket Link:

https://trinket.io/python3/b176b9eb81

Program top level design (pseudocode)

1.	Get new member first name, surname, category and password.	OUT: first name, surname, category, password
2.	Read existing member data from file to parallel arrays. Add new member data to parallel arrays. Display first name, surname and category of all members.	IN: first name, surname, category, password OUT: category()
3.	Find and display the number of members in each category and the total number of members.	IN: category()

Refinements

1.1	Get first name	
1.2	Get surname	
1.3	Get category	
1.4	Call function to get a valid password	OUT: password
1.4.1	Loop until password is valid	
1.4.2	Ask the user to enter a password	
1.4.3	Check that the first character is a capital letter (ASCII values 65 to 90)	
1.4.4	Check that the last character is #, $ or % (ASCII values 35 to 37)	
1.4.5	Return a valid password	
2.1	Read existing member data from file into four parallel arrays: firstName(), surname(), category(), password()	
2.2	Add the new member data to the existing member data in the parallel arrays	
2.3	Display the first name, surname and category of all members	

```
# 2. Read existing data from walkers file and add new member details and display all members
def readFileDate(newFirstName,newSurname,newCat,newPass):

    # Initialise Parallel arrays
    firstName = [""]*50
    surname = [""]*50
    category = [""]*50
    password = [""]*50

    # Read file data into arrays
    counter = 0
    with open("members.txt") as readFile:
        line = readFile.readline().rstrip("\n")
        while line:
            items = line.split(",")
            firstName[counter] = items[0]
            surname[counter] = items[1]
            category[counter] = items[2]
            password[counter] = items[3]

            line = readFile.readline().rstrip("\n")

            counter +=1
   #counter will equal the number of lines in the text file at the end of readFile
    # Add new members details to the end of populated arrays
    # counter = 10. Populating Arrays 0-10 will give 11 values so using counter will add on a new array item
    firstName[counter] = newFirstName
    surname[counter]  = newSurname
    category[counter]  = newCat
    password[counter]  = newPass

    # Display all the members details.
    print("**********************")
    print("Our member(s) are:")
    for loop in range(0,counter+1):  #+1 to ensure that the new member is also printed off.
        print(firstName[loop],surname[loop],category[loop])

    return category
```

```
Running: 2020 Higher  Solution - (parallel arays).py
Please enter the members first name: Alan
Please enter the members surname: Simpson
Please enter the members category: Adult
Please enter valid password: Adfesf%
**********************
Our member(s) are:
Angela Rich Adult
Siraj Adkins Junior
Stefano Love Senior
Cameron Wilder Junior
Griff Sutherland Adult
Amaan Sosa Senior
Isaak Schroeder Junior
Nana Galloway Junior
Lila Blanchard Adult
Eren Acosta Adult

Alan Simpson Adult
************************************
There are currently 4 members
There are currently 5 members
There are currently 2 members
Total current membership is 11
>>>
```

```python
#-----------------------------------------------------------------------
# 3. Find and display the number of members in each category.
def categoryCount(category):
    junior = 0
    adult = 0
    senior = 0

    # Count Occurence but with three counts
    for loop in range(len(category)):
        if category[loop] == "Junior":
            junior = junior + 1
        if category[loop] == "Adult":
            adult = adult + 1
        if category[loop] == "Senior":
            senior = senior + 1

    # Display results of count occurence
    print("************************************")
    print("There are currently",junior,"members")
    print("There are currently",adult,"members")
    print("There are currently",senior,"members")
    print("Total current membership is",junior+adult+senior)
#-----------------------------------------------
#Main Program
newFirstName = ""
newSurname = ""
newCat = ""
newPass = ""

# 1. Call function to get new member's details
newFirstName,newSurname,newCat,newPass = getMember()

# 2. Call procedure to read the data from the text file and add new member
category = readFileDate(newFirstName,newSurname,newCat,newPass)

# 3. Find the number in each category.
categoryCount(category)
```

-