1811ICT/2807ICT/7001ICT Programming Principles Workshop 8

School of Information and Communication Technology

Griffith University

|  |  |
| --- | --- |
| *Goals* | This workshop focusses on everything in the course up to files. |
| When | Week 9 |
| Marks | 3 |
| Due | There are no pre-workshop tasks this week.  Workshop programming problems by 11:59pm on 16 May |

# Before your workshop class:

* Read all of this document.
* Review the lecture notes sections 1 to 21.
* **There are no pre-workshop tasks this week**.

# Workshop activities

## Problem 1

*Problem:* Write a program with a *function* that given a list of numbers, rotate the numbers so the first number becomes the last, and the rest of the numbers move one position forward. Do the rotation iteratively until the list of numbers returns to its initial form.

Input a list: 1 2 3 4

[1, 2, 3, 4]

[2, 3, 4, 1]

[3, 4, 1, 2]

[4, 1, 2, 3]

[1, 2, 3, 4]

*Answer*: Copy your code in the space given below and insert screenshots of your program output for the following two scenarios:

* 4, 2, 4, 3, 3, 1, 2
* 1, 2, 1, 2

***Copy your code here***

*def rotation(lst):*

*print (lst)*

*for i in range(len(lst)):*

*var = lst[0]*

*del lst[0]*

*lst.insert(len(lst),var)*

*print (lst)*

*lst = [4, 2, 4, 3, 3, 1, 2]*

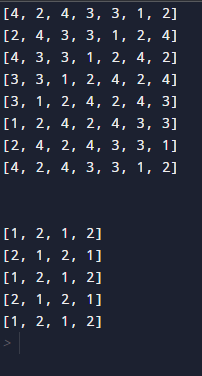
*lst1 = [1, 2, 1, 2]*

*rotation(lst)*

*print("\n")*

*rotation(lst1)*

***Insert your screenshots here***



## Problem 2

*Problem:* Given two lists, write a program with a *function* that takes these two lists as the input arguments and returns the list of all the elements in the first list that occur in the second list. The returned list shall not contain duplicate elements. Your main program will allow the user to enter two lists of numbers and end the program by inputting a blank line for list 1.

|  |
| --- |
| List 1: 1 3 3 6  List 2: 3 4 2 1 2 1 3  Output: [1, 3]  List 1: 3 4 2 1 2 1 3  List 2: 5 6 7 8  Output: []  List 1: |

*Answer*: Copy your code in the space given below and insert screenshots of your program output for the following two scenarios:

* List 1: 0 1 2 3 1 2 3 2 3 3

List 2: 2 4 6 1 3 5

* List 1: 1 1 2 2 3 3 4 4

List 2: 8 7 6 5 4 3 2 1 0

***Copy your code here***

*def same\_ele(list1, list2):*

*lst = []*

*for i in list1:*

*for j in list2:*

*if i == j and i not in lst: lst.append(i)*

*return lst*

*list1 = [0, 1, 2, 3, 1, 2, 3, 2, 3, 3]*

*list2=[2, 4, 6, 1, 3, 5]*

*print(same\_ele(list1,list2))*

*list1 = [1, 1, 2, 2, 3, 3, 4, 4]*

*list2 = [8, 7, 6, 5, 4, 3, 2, 1, 0]*

*print(same\_ele(list1,list2))*

***Insert your screenshots here***



## Problem 3

*Problem:* Write a program with a *function* that given a list of numbers, returns True if and only if all of the numbers in the list form an arithmetic progression, that is the difference between any two successive numbers in the list is the same. Your main program should read a file containing space-separated numbers as test lists, print each list and the outcome after calling the function.

File name: numbers.txt

[1 2 3 4] True

[10 20 30 40] True

[10 9 8 7] True

[2 7 8 3] False

[1 2 3 5] False

*Answer*: Copy your code in the space given below and insert screenshots of your program output for the following scenario:

* Use the file P3\_v1.txt as the source file.

***Copy your code here***

import os

os.chdir(r'D:\GU\Programming Principles\Workshop\WS08')

def arithmetic\_progression(arr):

    distant = -1

    for i in range(len(arr)-1):

        if distant ==-1: distant = abs(arr[i]-arr[i+1])

        elif abs(arr[i]-arr[i+1]) != distant: return False

    return True

def checknumber():

    file1 = open("P3\_v1.txt", 'r')

    while True:

        line = file1.readline()

        if line =="": break

        arr = [int(i) for i in list(line.split(" "))]

        print (arithmetic\_progression(arr))

    file1.close()

checknumber()

***Insert your screenshots here***

Text

Description automatically generated

## Problem 4

*Problem:* Given two lists, write a program with a *function* that merges these two lists in descending order. Your main program will allow the user to enter two lists of numbers and end the program by inputting a blank line for list 1. You are not allowed to concatenate the two lists into a new list and then call the built-in sort() function to sort the new list in descending order. But you are allowed to sort the two lists in descending order before merge. Don’t worry about the complexity of the merging algorithm.

List 1: 1 3 3 6

List 2: 1 4 5

[6, 5, 4, 3, 3, 1, 1]

List 1: 100

List 2: 1 1 3

[100, 3, 1, 1]

*Answer*: Copy your code in the space given below and insert screenshots of your program output for the following two scenarios:

* List 1: 0 1 2 3 1 2 3

List 2: 2 4 6

* List 1: 1 1 2 2 3 3 4 4

List 2: 8 7 6 5 4 3 2 1 0

***Copy your code here***

*lst1 = [1,1,2,2,3,3,4,4]*

*lst2 = [8,7,6,5,4,3,2,1,0]*

*lst1.sort()*

*lst2.sort()*

*lst1= lst1[::-1]*

*lst2= lst2[::-1]*

*lst = []*

*i = j =0*

*while True:*

*if i == len(lst1) or j == len(lst2):break*

*if lst1[i] == lst2[j]:*

*lst.append(lst1[i])*

*lst.append(lst2[j])*

*i+=1*

*j+=1*

*elif lst1[i] > lst2[j]:*

*lst.append(lst1[i])*

*i+=1*

*else:*

*lst.append(lst2[j])*

*j+=1*

*if i <len(lst1): lst += lst1[i:]*

*if j <len(lst2): lst += lst2[j:]*

*print (lst)*

***Insert your screenshots here***

**

**

## Problem 5

*Problem:* Book club members meet regularly for coffee at a **round cafe table**. Only a few members turn up randomly each time. It is a good thing that they keep records of who comes to each meeting, because one of them has been diagnosed with contagious book fever. It is very likely that if you sit next to someone with book fever, you will catch book fever.

The file meetings1.txt contains the first names of all the attendees at each meeting, in the order they **sat around the table**. It contains:

1. Chuck Trevor
2. Zack Olive Xander Ephraim
3. Ralph Wendy Ephraim Grace Leslie Phil Kathy
4. Binh Harry Ralph Xander Zack Chuck Uma Suzy Phil Kathy
5. Neville Leslie Kathy
6. Neville Harry\* Binh Vince Xander Zack Quisha Olive Phil
7. Yvonne Uma Trevor Fran Olive Phil Kathy
8. Harry Ralph Ephraim Denise Quisha Grace Phil
9. Binh Mandy Xander Ephraim Leslie Olive Fran
10. John Olive Chuck Mandy

Poor Harry, at meeting number 6, is marked with an asterisk because we know he was infected with book fever then.

Write a program that reports the **names and number of club members infected** up to each meeting from the first meeting where anyone was infected, like this:

|  |
| --- |
| Enter file name: meetings1.txt   1. Harry Neville Binh 3 2. Harry Neville Binh 3 3. Harry Neville Binh Ralph Phil 5 4. Harry Neville Binh Ralph Phil Mandy Fran 7 5. Harry Neville Binh Ralph Phil Mandy Fran John Chuck 9 |

The program should work in general for any such file with exactly one name marked with an asterisk. The file meetings2.txt should produce this output.

1. Phil Andrew Kathy 3
2. Phil Andrew Kathy 3
3. Phil Andrew Kathy Ephraim Harry 5
4. Phil Andrew Kathy Ephraim Harry 5
5. Phil Andrew Kathy Ephraim Harry 5
6. Phil Andrew Kathy Ephraim Harry 5
7. Phil Andrew Kathy Ephraim Harry Suzy Leslie 7

*Answer*: Copy your code in the space given below and insert screenshots of your program output for the following scenario:

* Use the file P5\_v1.txt as the source file.

***Copy your code here***

def fever\_next(line, lst):

    arr = line.split(' ')

    arr[-1] = (arr[-1])[:-1] *#remove \n*

    num = arr[0]

    del arr[0]

    if "\*" in line:

        if len(arr) == 1:

            line.remove("\*")

            return [line]

        elif len (arr) == 2 or len(arr) ==3:

            line.remove("\*")

            lst+= line.split(' ')

        else:

            for i in range(len(arr)):

                if '\*' in arr[i]:

                    position = i

                    arr[i] = (arr[i])[:-1]

                    break

            if position == len(arr) -1:

                lst += [arr[position], arr[position-1],arr[0]]

            elif position == 0:

                lst += [arr[position], arr[position+1], arr[-1]]

            else:

                lst += [arr[position], arr[position-1], arr[position+1]]

    else:

        base = lst.copy()

        for i in range(len(arr)):

            if arr[i] in base:

                if i == len(arr) -1:

                    if arr[i-1] not in lst: lst.append(arr[i-1])

                    if  arr[0]  not in lst: lst.append(arr[0])

                elif i == 0:

                    if arr[i+1] not in lst: lst.append(arr[i+1])

                    if  arr[i-1]  not in lst: lst.append(arr[i-1])

                else:

                    if arr[i-1] not in lst: lst.append(arr[i-1])

                    if  arr[i+1]  not in lst: lst.append(arr[i+1])

    return lst, num

def checknumber():

    file1 = open("P5\_v1.txt", 'r')

    fever = False

    lst = []

    while True:

        line = file1.readline()

        if line =="": break

        if '\*' not in line and fever != True: continue

        elif '\*' in line:

            fever  = True

        lst,num = fever\_next(line,lst)

        print(num, end = " ")

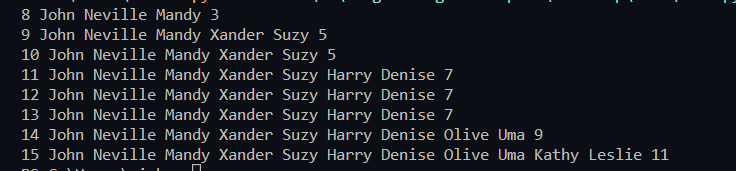
        print (\*lst, sep = " ", end = " ")

        print(len(lst))

    file1.close()

checknumber()

***Insert your screenshots here***



# Submission and marking

There are no pre-workshop tasks this week.

For workshop tasks, please submit this document with copied codes and inserted screenshots using the provided submission link in the course website. Students get 3 marks if they complete four or more problems correctly, or 2 marks if they complete three problems correctly, or 1 mark if they complete one or two problems correctly, or 0 marks without any attempt.