**Information and Computer Science Department**

**ICS 104: Introduction to Programming in Python and C**

**Final Exam, Term 213**

**Wednesday, August 10, 2022**

**Duration: 150 minutes**

**Name:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID:**  |  |  |  |  |  |  |  |  |  |

**Instructor Section: Select one**

|  |  |
| --- | --- |
| **Instructor** | **Section** |
| Dr ADAM SALAHADIN |  [ ] 1 (UMTW 8:10 – 9:00)  |
| Dr. WASFI AL-KHATIB |  [ ] 2 (UMTW 9:20 – 10:10) |
| Dr. MAHMOOD NIAZI |  [ ] 3 (UMTW 10:30 – 11:20)  [ ] 13 (UMTW 1:00 – 1:50) [ ] 14 (UMTW 12:00 – 12:50) |
| Dr. HUSNI AL-MUHTASEB |  [ ] 4 (UMTW 12:00 – 12:50)  [ ] 11 (UMTW 9:20 – 10:10) [ ] 12 (UMTW 11:00 – 11:50) |
| Dr. AHMAD AL-HERZ |  [ ] 5 (UMTW 13:00 – 13:50)  [ ] 7 (UMTW 9:20 – 10:10) [ ] 9 (UMTW 12:00 – 12:50) |
| Mr. YAHYA GAROUT |  [ ] 15 (UMTW 8:10 – 9:00) |

**Instructions**:

* Write your name, ID, and Section # on
	+ This page.
	+ The upper left corner of the green answer sheet.
* Bubble your ID, Section number and Test Code number on the green sheet, and make sure that the bubbles match with the numbers that you wrote/checked on this page.
* Write your name and ID number on the white Answer Sheet.
* **Parts 1 & 2.** Use the green sheet to answer the 40 multiple choice questions.
	+ Use HB2.5 pencils only.
	+ Use a good eraser. DO NOT use erasers attached to the pencil. If you erase a bubble, make sure that you do not leave any trace of penciling.
	+ When bubbling your answers, make sure that the bubbled space is fully covered.
* **Part 3.** Use the white answer sheet to answer the Code Writing questions.
* The exam is closed book and closed notes.
* Make sure that the exam consists of 15 pages (including this page).
* All types of calculators, pagers or mobile phones are not allowed during the exam.
* Make sure to turn off your mobile phone and keep it under your seat.

**Part 3. [ 40 Points] Code writing (Python)**

1. **[20 points]** Write python function def majority (theList) that takes as an argument a list of integers theList and returns a majority element and its frequency in the list. A majority element is the most frequent integer appearing in the list. Please note the following:
* If you provide the list [2, 4, 1, 0, 8, 2, 1, 5, 1] as an argument, it should return (1,3), where 1 is the majority element and 3 is the frequency of 1 in the list.
* A list may have more than one majority element. In that case, you can return either one of them. For example, if you provide the list [2, 4, 1, 0, 8, 2, 1, 5, 1, 2], your function may either return (1, 3) or (2, 3).
* You do not need to write the main program. Just the function.

Two possible solutions:

**Solution 1:**

def majority (theList) :

 b = theList.copy()

 b.sort()

 majorityElement=b[0]

 maxFrequency=1

 tempFrequency=1

 i=1

 while i < len(b) :

 if b[i] == b[i-1] :

 tempFrequency += 1

 elif tempFrequency > maxFrequency :

 maxFrequency = tempFrequency

 majorityElement = b[i-1]

 tempFrequency = 1

 i += 1

 if tempFrequency > maxFrequency :

 return (b[i-1],tempFrequency)

 else :

 return (majorityElement,maxFrequency)

**Solution 2:**

def majority (a) :

 maxFrequency=0

 for i in range(0, len(a)) :

 tempFrequency = 0

 for j in range(i, len(a)) :

 if a[i] == a[j] :

 tempFrequency += 1

 if tempFrequency > maxFrequency :

 maxFrequency = tempFrequency

 majorityElement = a[i]

 return (majorityElement,maxFrequency)

1. **[20 points]** Write a python program that checks whether the values in file "final.txt" is in increasing order or not.

Note:

* The file has at least two integer values.
* Each line has only one value.

For example:

If the file "final.txt" contains:

1

5

5

7

Then it is ordered, and

If the file "final.txt" contains:

5

1

3

Then it is not ordered

**Possible Solution**

inFile = open("final.txt", "r")

values = []

for line in inFile:

    values.append(int(line))

inFile.close()

isOrdered = True

i = 0

end = len(values) - 2

while isOrdered and i <= end:

    if values[i] > values[i + 1]:

        isOrdered = False

    else:

        i = i + 1

if isOrdered:

    print("values in final.txt are ordered")

else:

    print("values in final.txt are not ordered")