广东以色列理工学院
Guangdong Technion-Israel Institute of Technology

234128 – Introduction To Computing With Python

Winter Semester 2020-2021

HW2

In charge : Israel Gutter   Israel.gutter@gtiit.edu.cn

Due date : Monday, 2021-01-11 23:59

Mission 1 : (hw2q1.py)

Write the following functions in the file hw2q1.py :

def sort_2(a):

a is a list of 2 int numbers exactly. If a[0] > a[1] then the function swap between the 2 elements. The function returns the sorted list as a result.

Requirements :

Must NOT use any other function.

def sort_3(a):

a is a list of 3 int numbers exactly. The function sorts a from left to right, small to big.

It returns the sorted list as a result.

Requirements :

All operations upon a should be done by using sort_2 ONLY. The code of sort_3 must NOT have nothing else but calling to sort_3.

def sort_n(a):

a is a list of n int numbers exactly. n>=0.

The function sorts a from left to right, small to big.

It returns the sorted list as a result.

Requirements :

All operations upon the list a should be done by using sort_3 ONLY. The code of sort_n

can include loop and/or if.

Remark : Notice that the list can be of ANY length. Even empty.

In case you can not solve without calling to sort_2, then you can also call to sort_2, but try to

do this as minimum times as you can.

Write a main code that will get from input any number of int numbers (1 line of input. the numbers come one after other. Between every 2 numbers appears 1 or more space. Spaces can appear also in start/end of the input line.

The program displays the input numbers as a sorted list, left to right, small to big.

Check your program upon the given examples. Verify that the format of input is correct. Specifically check about spaces/new lines. Use input/output file redirection. Use cmp_files.py.

Mission 2 :

In this mission we'll deal with a mountain.

As we've discussed in class, a mountain is a list in which there is any index i such that all the elements starting with the first element till a[i] (including a[i]) are "going up", which means a[j]<=a[j+1] for all 0<=j<i. Then for all the elements starting with a[i] till the last element (including the last element), these elements are "going down", which means a[j]>=a[j+1] for all i<=j<n (n is length of the mountain).

Not necessarily we have elements left to i. Not necessarily we have elements right to i.

A list that has no more than 2 elements is a mountain.

Part A : (hw2q2a.py)

Write a function

def even_values(a):

a is a mountain. The function returns the number of different **even** values (not elements) that appear in the mountain.

Requirements : The function should be efficient.

Write a main code that will get from input any number of int numbers (1 line of input. the numbers come one after other. Between every 2 numbers appears 1 or more space. Spaces can appear also in start/end of the input line. Assume that the input is a legal mountain.

The program displays the number of different even values that appear in the mountain.

Check your program upon the given examples. Verify that the format of input is correct. Specifically check about spaces/new lines. Use input/output file redirection. Use cmp_files.py.

Part B : (hw2q2b.py)

Write a function

def max_even(a):

a is a mountain in which all elements are different one of the other. The function returns an index in which there appears the max value of the mountain. If the list is empty then return 0. (Notice that also when having only 1 element, 0 is returned).

Requirements : The function should be efficient.

Write a main code that will get from input any number of int numbers (1 line of input. the numbers come one after other. Between every 2 numbers appears 1 or more space. Spaces can appear also in start/end of the input line. Assume that the input is a legal mountain.

The program displays the max value of the mountain. If no max value, the program displays a suitable message.

Check your program upon the given examples. Verify that the format of input is correct. Specifically check about spaces/new lines. Use input/output file redirection. Use cmp_files.py.

Mission 3 : (hw2q3.py)

We have the following code of the function quicksort :

```
def quicksort(a):
    if len(a)<2:
        return a
    id=partition(a)
    a[:id]=quicksort(a[:id])
    a[id:]=quicksort(a[id:])
    return a
```

Write the function

```
def partition(a):
```

Requirements : `Time Θ(len(a))   Space Θ(1) in place`

Use only one loop (without recursion).

Notice that the function partition should divide the list a into 2 NON EMPTY parts such that all elements in the left part are small or equal than all elements in the right part.

Write a program (main code) that will get from input any number of int numbers (1 line of input. the numbers come one after other. Between every 2 numbers appears 1 or more space. Spaces can appear also in start/end of the input line.

The program displays a list of all input numbers sorted from left to right, big to small.

The sorting should be done by using quicksort.

Check your program upon the given examples. Verify that the format of input is correct. Specifically check about spaces/new lines. Use input/output file redirection. Use cmp_files.py.

Submission :

Make a zip file whose name is <id>.zip whereas <id> is your id (9 digits exactly).
Example – 999003645.zip

Please verify – ONLY zip file. NOT rar file or any other format.

The zip file should contain the following files :

hw2q1.py   hw2q2a.py   hw2q2b.py   hw2q3.py   student.txt

NOTHING ELSE should be included in the zip file. NO any sub folder should appear in the zip file.

student.txt should contain your personal details – ENGLISH ONLY. Fill in the file that is given.

DO NOT CHANGE WHAT IS ALREADY WRITTEN. DO NOT CHANGE THE NAME OF THE FILE. Just

complete the missing details.

Submit the zip file in Moodle.

Good Luck !