Problem Set 2

Handed out: Tuesday (6/5/2019)

Due: Tuesday (12/5/2019)

This problem set will introduce you to defining and calling functions while passing arrays as arguments. This problem set has a single problem which you should save as Your_10_digit_sudent_id.c, e.g. if a student's id is 2013-3-60-23 his code should be saved as 2013-3-60-23.c

Collaboration

You may work with other students and share ideas. But in no circumstances you are allowed to share your code with others. If a case arises as such both the provider and receiver will be treated equally guilty and get a zero.

Before you start: Read the style guide

Read the style guide section from the link provided in the course site.

Problem Description

You will be given a 1d array. You have to find if the given array is a **Mountain** array or not using the following instructions.

A Mountain array is an array where the elements form a mountain - first, they increase to a peak and then they start to decrease to the end of the array.

Consider the following array, whose peak is 30 (The largest element of the array). The elements from the beginning to the peak are 1, 5, 10, 15, 16, and 30, which are gradually increasing, also expressed as monotone increasing. The elements from the peak to the end of the array are 30, 27, 25, 24, 20, 10, 7, 6 and 6 which are gradually decreasing, also expressed as monotone decreasing. Hence, the array is called a Mountain array.

1	5	10	15	16	30	27	25	24	20	10	7	6	6

Along with the problem set, we have provided you with a `main.c` file which contains two user-defined functions and a main function. Your job is to complete those 3 functions. For completing this problem set, you have to use those pre-defined functions. In no way you are allowed to use any of your own user-defined functions.

A. Getting Started

The problem can be divided into four parts.

i) Find the index (l) of the largest element of the input array. Complete find_max function

ii) Check if the sub-array from index 0 to l is monotone increasing or not. Complete is_monotone_increasing function

iii) Check if the sub-array from index l to $size - 1$ is monotone decreasing or not. Complete is_monotone_decreasing function

iv) Based on the output of (ii) and (iii) decide whether it is a mountain array or not. Write additional codes in main function

B. int find_max(int a[], int size)

This is a pretty straightforward function which expects 2 parameters – an integer array, the size of it. And it will return the index of the max element of the array - an integer. In case, the maximum element is a duplicate, pick the first occurrence.

C. int is_monotone_increasing(int a[], int size)

This function expects 2 parameters – an integer array, the size of it. And it will return 1 if a is monotone increasing, else 0.

An array is called monotone increasing if all the elements are sorted in an increasing manner, that is each element is either less than or equal than the next element.

Consider the array1, where each element is sorted in an increasing manner, that is $1 < 5 < 10 < 15 < 16 < 30$

That's why it is monotone increasing. Whereas in array 2, $1 < 5 < 10 \nleq 8$ ... . Hence, array2 is not monotone increasing.

1	5	10	15	16	30

array1

1	5	10	8	16	30

array2

D. int is_monotone_decreasing(int a[], int size)

This function expects 2 parameters – an integer array, the size of it. And it will return 1 if a is monotone decreasing, else 0.

An array is called monotone decreasing if all the elements are sorted in a decreasing order, which is each element is either greater or equal than the next element.

Consider the array1, where each element is sorted in an increasing manner, that is $30 > 27 > 25 > 24 > 20 > 10 > 7 > 6 \geq 6$

That's why it is monotone decreasing.

30	27	25	24	20	10	7	6	6

Sample Input-Output

Input1
Enter array size: 14
Enter array: 1 5 10 15 16 30 27 25 24 20 10 7 6 6
Output1
Mountain array

Input2
Enter array size: 14
Enter array: 1 15 10 5 16 30 27 25 24 20 10 7 6 6
Output2
Not Mountain array

Input3
Enter array size: 16
Enter array: 1 5 10 15 16 30 30 30 27 25 24 20 10 7 6 6
Output3
Mountain array

Carpe Diem!