COMP 132: Advanced Programming Spring 2019

Problem Session 10

Arrays in C

Question 1: (Intersection of Sets) Use one-dimensional arrays to solve the following problem. Read in two sets of numbers, each having 10 numbers (use predefined size variable). After reading all values, display the unique elements common to both sets of numbers.

Pointers in C

Question 2: In this question, you are asked to develop a C program that transforms an unordered array into an alternating derivative sequence (ADS).

An ADS is a sequence that decreases and then increases repeatedly like a sinusoidal wave. More formally, Let A be a sequence of length n. Then, A is an ADS iff $A[i] \geq A[i+1]$ for all even $0 \leq i < n - 1$ and $A[i] \leq A[i+1]$ for all odd $0 \leq i < n - 1$.

For instance, $A = 8, 7, 10, 1, 3, 2$ is an ADS since

$A[0] = 8 \geq 7 = A[1]$,

$A[2] = 10 \geq 1 = A[3]$,

$A[4] = 3 \geq 2 = A[5]$ and

$A[1] = 7 \leq 10 = A[2]$,

$A[3] = 1 \leq 3 = A[4]$.

In this PS, you will be given an unordered array arr (hardcoded in the main method) and transform it to an ADS preserving the elements in it. You will implement and use the following methods for this purpose:

a) int* findMinLocation(int* start, int* end): Scans the interval in an array beginning from (including) start and ending at (excluding) end pointers and returns a pointer between those two such that the value at the return location is the minimum in the interval.
b) int* findMaxLocation(int* start, int* end): Scans the interval in an array beginning from (including) start and ending at (excluding) end pointers and returns a pointer between those two such that the value at the return location is the maximum in the interval.

c) void swap (int* p1, int *p2): Takes two pointers and swaps the values in these locations.

Then, you will iterate over the array in main and do the following at iteration i:

a) Find the position $pMax$ that is the location of the maximum element in the interval $[2i, n)$.

b) Swap the elements at locations $pMax$ and $2i$.

c) Find the position $pMin$ that is the location of the minimum element in the interval $[2i + 1, n)$.

d) Swap the elements at locations $pMin$ and $2i + 1$.

Note that you will obtain an ADS prefix for the interval $[0, 2i]$ after i^{th} iteration of this procedure.

For initial $arr = 3, 9, 4, 12, 48, 1, 5$, the program should produce the following arrays after each iteration:

After iteration 1: 48 1 4 12 3 9 5

After iteration 2: 48 1 12 3 4 9 5

After iteration 3: 48 1 12 3 9 4 5