One Dimensional Array – Lab Assignments

(To be submitted in AUMS before 13th 11 pm)

Use functions for all the below programs. Follow the coding standards. Use meaningful name for variables and functions. Each subtask can be defined as a function. Pass the input required as parameters and return the result. Give comments for each section

DO NOT COPY FROM INTERNET.
We will be doing a plagiarism check during evaluation

1. Write a program to interchange the largest and the smallest number in the array.
2. Print the sum of even positioned numbers in the list
3. Find the kth smallest element in the Array
4. Write a program to sort an array of elements using Bubble sort.
5. Given an unsorted array and two numbers x and k, find k closest values to x
6. Read a value k from the user and using k do k left rotation and right rotation depending on the user choice Left or Right, print the rotated value

   Sample input Array : 1 2 3 4 5

   Output after one left rotation : 2 3 4 5 1

   Output after 2 left rotations: 3 4 5 1 2

   Output after 1 right rotation: 5 1 2 3 4
7. Count all pairs whose sum is an odd number

   Input: a[] = {9, 14, 6, 2, 11}, b[] = {8, 4, 7, 20}
   Output: 9
   {9,8}{9,4}{9,20},{14,7},{6,7},{2,7}, {11, 8},{11,4},{11,20},

   Input: a[] = {2, 4, 6}, b[] = {8, 10, 12}
   Output: 0
8. Given two sorted lists A and B, Create a new sorted array with maximum possible length, which alternatively takes elements from both the lists. Let the first element be from list A. (Do not do sorting after creation of the output array. Pick the elements alternatively such that array is sorted at the end of picking)

   Input: a[]={20,34,45,50} b[]={1,5,25,40,90}

   Output c[]={20, 25,34,40,45,90}

   [While picking alternate elements,we couldn't pick 1 ,5 since 20 is picked and it is greater than 1 and 5]

9. Using arrays help a Car service centre serve cars based on a QUEUE which follows a First Come first Serve order.. As the customers walk in, they register (put ticket) their vehicle no into the array. At any time admin at Service centre should be able to get the ticket no to be serviced NOW which should strictly follow the FCFS strategy. Once the service centre completes the service of a vehicle, they should have option to delete the ticket from the QUEUE. Note that array will result in a QUEUE data structure which follows a FIRST IN FIRST OUT (FIFO) strategy.

   Functions to be implemented:
   - REGISTER (Incoming customer should register with car no)
   - WHONEXT (At any time admin should know which car is waiting next in the QUEUE)
   - DELETE (Once service is completed, admin should delete the car from the QUEUE)
   - DISPLAYQUEUE (At point of time queue should be displayed in FCSF order

   In one run of the program, the user should be able to call the above functions as many times as he/she wants. Do a choice based iteration.

   Eg: QUEUE STATUS : 123 567 111 999

   WHONEXT → 123

   INSERT(333)--→ inserted 333 at the end of queue

   DISPLAYQUEUE -> 123 567 111 999 333

   DELETE()-→. Deleted element is 123

   DISPLAYQUEUE → 567 111 999 333

10. Imagine you are maintaining a deck of cards in LAST –IN-FIRST OUT fashion. Any card which is inserted at the end will come out first. Do a choice based iteration of the following functions.

    INSERT: Should be able to insert into the DECK.
    DISPLAYDECK: should display the status of deck in the order
    DELETE: Should delete the lastly inserted element.
    PEEP: At any time, PEEP should display the next element in the DECK

    Eg: DECK STATUS :1 4 25 6 78
    INSERT(20)-> inserted 20 at the front side
    DISPLAYDECK→ 20 1 4 25 6 78
    PEEP() ---→ element seen when peeped in is 20
    DELETE()→ deleted element is 20
    DISPLAYDECK-> 1 4 25 6 78