

PDS LAB [Date: 1st Oct 2024]
Assignment – 5 [Pointers & DMA]

Instructions:

1. Create a directory named as Lab-5.
2. Give the name of the program as <p>.c where <p> implies the problem number, like 1.c, 2.c, 3.c, etc. Store all the programs of this week under this directory.
3. You should upload all .c files (1.c, 2.c, 3.c) to the Moodle course web page latest by 5.00 PM (without penalty). The cutoff time will be till 5.15 PM with a penalty of 25% on your secured marks (i.e., if you secured 80 marks, after penalty you will get 60 marks). Beyond 5.15 PM, the moodle system will not allow you to submit, as a result you will get zero.

Note: For solving this assignment don't use static arrays. Use only pointers and DMA to solve the problems. Don't use array notation (i.e., []) for accessing the elements of the array.

p1] Write a C program to perform the following: **[30 Marks]**

- (a) Capture the array x of real numbers using a C function **input_arr()**, and the size of the array x will be specified by the user during run-time, before calling this function.
- (b) Once, the array is filled up, compute the mean (M) of the array using **float* mean_arr()** function .
- (c) Insert mean (M) value using C function **insert_mean_arr()** prior to the array element such that the absolute difference between the inserted value and array element is minimum. Before, calling this function, the array size to be resized as per the requirement. Don't use any additional arrays.
- (d) Finally, display array elements using **display_arr()** function.

Example: Array x = 1 2 3 4 5

Mean (M) = 3.0

Array after insertion of Mean = 1 2 3 3 4 5

p2] Write a C program to perform the following:

[35 Marks]

- (a) Input a string using a C function ***input_str()***, whose length is specified by the user during run-time, before calling this function.
- (b) Split the input string into four strings namely `rise_str`, `fall_str`, `rise_fall_str` and `flat_str`, using four C functions ***rise_str()***, ***fall_str()***, ***rise_fall_str()*** and ***flat_str()*** based on the following criteria. The size of the array to be resized dynamically based on the size of the identified word. You may declare pointers globally for this task, if required.
 - (i) `rise_str` consisting of words whose ascii values of the sequence of characters within the word follow rising (increasing) pattern. Ex: `aerv`, `egjp`, `Pmsy`
 - (ii) `fall_str` consisting of words whose ascii values of the sequence of characters within the word follow falling (decreasing) pattern. Ex: `faX`, `vtdb`
 - (iii) `rise_fall_str` consisting of words whose ascii values of the sequence of characters within the word follow rise-fall (up-down) pattern. Ex: `ksr`, `pds`
 - (iv) `flat_str` consisting of words whose ascii values of the sequence of characters within the word follow flat (remains same) pattern. Ex: `ssss`, `MMMM`, `a`
- (c) Display the resultant strings using ***display_str()*** function.

Don't use any additional pointers. You may need five pointers for this task.

Example:

Input string : `abcd xxxxx zxto ccccc cat dog pqrst fedc`

`rise_str` : `abcd pqrst`

`fall_str` : `zxto fedc`

`rise_fall_str` : `cat dog`

`flat_str` : `xxxxx ccccc`

p3] Write a C program to perform the following:

[35 Marks]

- (a) Input a string using a C function ***input_str()***, whose length is specified by the user during run-time, before calling this function.
- (b) Determine the set of elements from the given string using ***str_set()*** function, based on the property of set (determine the unique elements by removing the duplicates).
- (c) After entering two strings and determining their respective sets, perform set operations using user defined C functions ***set_union()*** and ***set_intersection()***.
- (d) Finally, use ***display()*** function to display individual sets and union and intersection of the resultant two sets.

Don't use any additional pointers. You may need six pointers for this task.

String-1 : PDS Lab on pointers

String-2 : PDS Theory on strings

Set-1 : PDS Labonpitters

Set-2 : PDS Theorystig

Union of Set-1 and Set-2 : PDS LabonpittersThyg

Intersection of Set-1 and Set-2 : PDS eornits

THE END