MCA-II
Advanced Programming
Assignment-II
2020

Note : (1) Write modular programs (using functions)
(2) Do not use global variables
(3) Use proper variable names
(4) Program should contain program definition as comments and should have popper indentation

Pointers
---

1. Write a program using pointers to read array of integers and print its elements in reverse order.
2. Write a program using pointers to find minimum and maximum element of an array and display it along with the address at which it is located.
3. Write a program to count the number of vowels, consonants, digits and white space characters using pointers.
4. Write a program using pointers to implement the transpose of a matrix.
5. Write a program using pointers to implement the matrix multiplication.
6. Write a program to perform summation of a matrix using pointers.
7. Write a program to sort the list of strings using pointers.
8. Write function that receives a sorted array of integers and an integer value, and inserts the value in correct place.
9. Write a function that will round a floating point number to an indicated decimal place eg: The number 17.457 would yield the value 17.46 when it is rounded off to two decimal places.
10. Write a function using pointers to exchange the value stored in two locations in the memory.
11. Write a C functions using pointer and character array to implement the following
    (a) Find the first occurrence of a character in the given string. The function should return the position in the string.
    (b) Find the first occurrence of a string in another string. The function should return the position in the string.
    (c) Delete all occurrences of a character from a string.
    (d) Delete all occurrences of a string from another string.
    (e) Delete all occurrences of a character from a string. Ignore Case.

(f) Delete all occurrences of a string from another string. Ignore Case.

(g) Copy one string to another string.

(h) Copy n characters of one string to another string.

(i) Find length of the string and toggle the characters of the string.

(j) Convert string to all upper case.

(k) Convert string to all lower case.

(l) Sort an array of string.

(m) Append one string to another string.

(n) Append at most n characters of one string S2 to another string S1.

(o) Reverse all the characters in the string.

(p) Compare two strings S1 and S2. The function should return -1, 0 or 1 if S1 < S2, S1 = S2 and S1 > S2 respectively.

(q) Compare two strings S1 and S2. The function should return -1, 0 or 1 if S1 < S2, S1 = S2 and S1 > S2 respectively.
Ignore case.

(r) Compare at most n characters of two strings S1 and S2. The function should return -1, 0 or 1 if S1 < S2, S1 = S2 and
S1 > S2 respectively.

(s) Compare at most n characters of two strings S1 and S2. The function should return -1, 0 or 1 if S1 < S2, S1 = S2 and
S1 > S2 respectively.Ignore case.