10504: Computer Programming
Lab Assignment 4

Swarnim Soni

1. **Accessing Arrays**

   (a) WAP to print an input string in reverse order.

   (b) WAP to replace all characters with another character, take the string, character to replace and the character to substitute from user. e.g. string = ”Hello World”, character to replace = ’o’, replacing character = ’X’, final string = ”HellX WXrld”

   (c) *Challenge:* WAP to remove a character from the entire string. e.g. if string is “how are you”, and character ’o’ is removed, final string will be ”hw are yu”

   Note: using library functions from ”strings.h” is not allowed.

2. **Permutation**

   (a) WAP to take an integer from user and generate its next permutation,

   (b) Modify above part a), that generates all permutations of the number from the begining. (You may need sorting for this problem).

   (c) *Challenge*: can you generate all permutations of smaller length? e.g. take number 123, all permutations of length 2 are 12,13,21,23,31,32. Take the number (like 123 in this example) and length of permutations (like 2 in this example) from user.

3. **Palindromes**

   (a) WAP that takes a string from user and prints out if the string is palindrome or not. Examples of palindrome are ”1213121”,”hannah”. Palindromes are strings with a property that they read the same forwards and backwards.

   (b) WAP that takes a string from user and find out it longest palindromic substring.

   (c) *Challenge:* A string is “almost palindrome” if by removing exactly 1 character from it, the string becomes palindrome. e.g. “acbc” is almost palindrome because removing first ’a’, string becomes palindrome (”cbc”). And ”abcd” is not an ”almosrt palindrome” because it can not be made a palindrome by removing exactly one character from it.

4. Randomly generate an array of 1000 elements, each element has values from 0 to 100 (0 and 100 inclusive). You need to write following:

   (a) a function "**float findMean(int* inputArray, int n)**" that takes the array as input and returns mean of all the elements in the array

   (b) a function "**float findMedian(int* inputArray, int n)**" that takes the array as input and returns median of all the elements in the array. For this task you may need to sort the array.

   (c) a function "**int** findFreqDist(int* inputArray, int n, int** outputArray)**" that takes an array (inputArray) as input and returns frequency distribution of all the elements in the array. For this task you need to create a 2D array of size 101 by 2 (outputArray). Each row will be a pair of (number, count of that number). Since there are only 101 elements, therefore the size of the array is 101 by 2.

   Note: 'n' above is size of input array.

5. **Matrices**

   (a) WAP to multiply matrices of size m * n and n* p. Initialize the two matrices in the code itself. (since taking input from the user is time consuming).

   (b) *Challenge*: WAP to compute determinant of a square matrix, initialize the matrix in the code itself.
   Assume all entries in the matrix are integers for this assignment, however, in solving real world problems you will encounter non-integer matrix.

   (c) *Challenge*: WAP that displays randomly generates a matrix of M by N (where M,N > 2), and then displays its elements in a spiral. e.g. if the matrix is

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

   then the output of the program should be 1 2 3 6 9 8 7 4 5