19CSE102 Computer Programming
Strings

1. Write a C program to find the frequency of a given character in a string.

2. Given a string, write a C program that counts the number of vowels, consonants, digits and whitespaces in a given string.

3. Write a C program that reverses a given string. Try a recursive solution.

4. Write a C program that removes all characters in a string except alphabets.

5. Given a string, write a C program that converts all uppercase characters in the string to lowercase and vice versa. (Note include `ctype.h` and use `isupper`, `islower` to test the case of a character and use `tolower` `toupper` functions.)

6. Write a C program to check whether a given string is a palindrome.

7. Write a C program to check the validity of a given password. The password validity can be checked based on the following policies.
   a) Password should not contain any space
   b) password should contain at least one digit
   c) password length should be 8 – 15 characters
   d) password should contain at least one lower case character
   e) password should contain at least one upper case character
   f) password should contain at leat one special character (@, #, $, %, !, & etc.)

8. Write a C program to check whether a given string of alphanumeric characters contain all digits from 0 to 9.

Do you also notice that transpose of a distance matrix is again the same distance matrix? Either modify your existing program or write another program or function to this too. (which means your program should transpose a matrix and should check the elementwise equality of two matrices!!)

9. Given a 2D array containing marks of *n* students in *m* subjects, write a C program to compute the following.
   a) Find *maximum* and *minimum* marks scored by any student in each subject.
   b) Find *maximum* and *minimum* marks scored by each student in any subject.
   c) Find the *average* marks scored by each student and *average* marks scored in each subject.
   d) Print the complete students vs grades matrix.

10. Given a character matrix with three possible characters *X, E* and *B* which represents the following

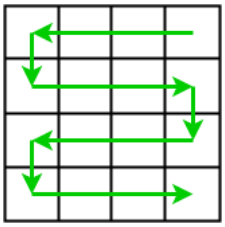
X – war zone
E – enemies
B – bombs
write a C program to compute whether all the enemies are killed (print *yes)* or not (print *no*) based on the following condition.

a) Bomb *B* can blast only in horizontal and vertical directions from one end to another.

For example
XXEX prints *yes*
XBXX
XEXX
XXBX

XXEX prints *no*
XBXX
XEXX
XXXX

11. Given a *n* X *n 2D* array print the elements of the array in *snake* fashion as shown in the figure below.



For example given the following matrix
1 2 3
4 5 6
7 8 9
the output should be
3 2 1 4 5 6 7 8 9