Problem Set 7

1. Write a menu-driven program to implement the following functionality.

    1. Enter a new char string
    2. Print the string
    3. Find length of the string
    4. Reverse the string
    5. Check if the string is a palindrome
    6. Count number of times letter 'e' appears in the string
    7. Print a histogram of all characters in the string (Problem 7 in Problem Set 6)
    8. Exit the program

2. Write a program to read two strings s1 and s2, and print how many times s2 appears in s1. For instance, if the inputs are *abracadabra* and *ab*, the output should be 2; if the inputs are *abababab* and *aba*, the output should be 3.

3. Write a program to read two integer arrays a1 and a2 which are already sorted in ascending order, and create a third sorted array a3. For instance, if a1 is {1, 3, 5, 7, 8} and a2 is {4, 4, 8}, the output should be {1, 3, 4, 4, 5, 7, 8, 8}.

4. What is the output of the following program?

```
#include <stdio.h>
#define _ []
#define __ "Code Obfuscation.\n"
#define _p printf
#define p_ const
#define _p_ char

int main() {
        p_ _p_ p _ = __;
        _p(p);
}
```

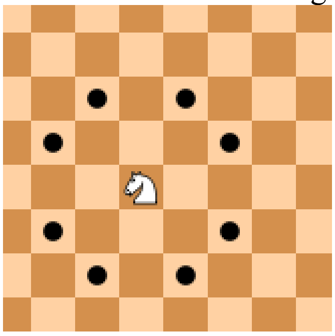
5. Similar to Problem 1 above, create a menu-driven program to (i) read N*N matrix of integers, (ii) print the matrix, (iii) square the matrix, (iv) find determinant of the matrix, (v) compute matrix-raised-to-k where k is read as an integer input, (vi) check if the matrix is lower-triangular, (vii) check if the matrix is diagonal, (viii) exit.

6. Read, store and print information about students using a 5-dimensional array. The dimensions are for Department (0=AE, 1=BT, 2=CE, 3=CS, 4=EE, 5=ME), Program (0=BTech, 1=DD, 2=MTech, 3=MS, 4=PhD), Year (2010 – 2016), Hostel (0=Cauvery, 1=Ganga, 2=Godavari, 3=Pampa, 4=Sarayu, 5=Tunga), and FavoriteEatery (0=CCD, 1=Gurunath, 2=IRCTC, 3=Zaitoon, 4=CampusCafe, 5=Himalaya). Thus, read Department, Program, Year, Hostel, FavoriteEatery for each student; store in an array; and in the end, print it for each student.

7. Given a two-dimensional matrix, print all its sentinel elements. We call an element a sentinel if it is the largest in its row but the smallest in its column. For instance, in the following matrix, matrix[3][2], that is, value 9 is the sentinel.

```
5   3    9  17
22  4  11  6
4  22  10  9
0   3    9   1
```

8. Create an 8x8 matrix resembling a chessboard. Assume that a knight (the piece that moves two-and-a-half places) is at matrix[0][0]. From any place, a knight can move to at maximum eight other places on the chessboard. Write a program to simulate knight's moves (print matrix after each move). Can you make sure that the knight moves to all the 64 cells without going to a cell twice?



9. A die has six faces with values 1 through 6. Assume you are rolling two dice, and storing the outcome frequency in an integer matrix[6][6]. Thus, if the dice outcome is (3, 6), we increment matrix[3-1][6-1]++. Write a program to roll two dice repeatedly 1000 times and note the frequencies in the matrix. In the end, print the matrix. You can use rand() function to generate a random number (see rand() example below).

```
#include <stdio.h>
int main() {
        srand(time(0));
        printf("%d\n", rand() % 6);
}
```