Labsheet 7

1. We have two arrays A and B, each with 10 integers. Write a program that takes the two arrays and their size as parameters and tests if every element of array A is equal to its corresponding element in array B.

2. Write a menu driven C program to do following operation on two dimensional array A of size m x n. The options are:

    - To input elements into matrix of size m x n
    - To display elements of matrix of size m x n
    - Sum of all elements of matrix of size m x n
    - To display row-wise sum of matrix of size m x n
    - To display column-wise sum of matrix of size m x n
    - To create transpose of matrix B of size n x m

3. Your friend sends an encrypted secret message to you. You know that the message is a single word and to read the original message you need to add a key to each character. Both of you know that the key is the ASCII difference of the first and last character of the message sent to you. Write a program to decrypt the message.

4. Write a program in C which accepts a 2D array of integers and its size as arguments and displays the elements of middle row and the elements of middle column. [Assuming the 2D Array to be a square matrix with odd dimension i.e. 3x3, 5x5, 7x7 etc...] Example, if the array contents is

    3 5 4
    7 6 9
    2 1 8

    Output through the function should be :
    Middle Row : 7 6 9
    Middle column : 5 6 1

5. Write a program to find the frequency of a particular element in an array. Read the element from the user in the main function. For example if the array contains the following elements : {1, 4, 3, 4, 7, 4, 3}, and the user entered 4, then the output would be: ***4 is repeated 3 times.***

6. Write your own C program that stores real matrix and finds the sum of elements from every column and product of elements from every row. Program prints the smallest sum (including parent column's index), and biggest product (including parent row's index). Sums and products should be stored in one-dimensional arrays.

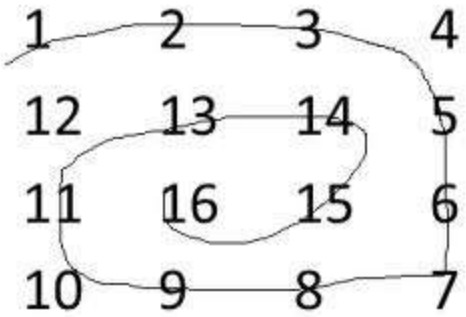
7. Write a C program that reads through real matrix, and finds the smallest element in main diagonal and smallest element in secondary diagonal.

8. Write a C Program for square matrix to calculate

    Left diagonal sum

Right diagonal sum

9. Write a C program to multiply two multiplication compatible matrices. Two matrices are said to be multiplication compatible if the number of columns of the first matrix and number of rows of second matrix are same.
10. Write a program to find the determinant of a 3X3 matrix.
11. Given a nxn square matrix. Write a code to print it in spiral order.



12. Write a C program to create a 3x3 magic matrix from any 9 consecutive positive numbers.
(Hint: Magic Square of size 3

---

2 7 6
9 5 1
4 3 8

Sum in each row & each column = 3*(3^2+1)/2 = 15

Magic Square of size 5

---

9 3 22 16 15
2 21 20 14 8
25 19 13 7 1
18 12 6 5 24
11 10 4 23 17

Sum in each row & each column = 5*(5^2+1)/2 = 65

Magic Square of size 7

---

20 12 4 45 37 29 28
11 3 44 36 35 27 19
2 43 42 34 26 18 10
49 41 33 25 17 9 1
40 32 24 16 8 7 48
31 23 15 14 6 47 39
22 21 13 5 46 38 30

Sum in each row & each column = 7*(7^2+1)/2 = 175

13. A binary matrix is an mxn matrix in which the elements are either 1 or 0. Given a binary matrix B, write a program to compute the simple run length encoding of each row of B.

14. Write a C program to read a sentence from the user and count the number of words in the sentence.

15. Given a sequence of characters, print consecutive sequence of characters in a line, otherwise print it in a new line.

    Example:

    Input :ABCDXYZACCD

    Output:ABCD

    XYZ

    A

    C

    CD

16. Given two strings, the task is to find if they are only less than or equal to k edit distance apart. It means that strings are only k edit distance apart when there are only k mismatches. Print Yes if there are less than or equal to k mismatches, Else No. Also print yes if both strings are already same.

Predict the output for the following programs.

9.
```
#include<stdio.h>
  int main()
        {
              char str1[] = "GeeksQuiz";
              char str2[] = {'G', 'e', 'e', 'k', 's', 'Q', 'u', 'i', 'z'};
              int n1 = sizeof(str1)/sizeof(str1[0]);
               int n2 = sizeof(str2)/sizeof(str2[0]);
              printf("n1 = %d, n2 = %d", n1, n2);
              return 0;
}
```

10.
```
#include<stdio.h>
 int main()
{
        char str[] = "%d %c", arr[] = "GeeksQuiz";
        printf(str, 0[arr], 2[arr + 3]);
        return 0; }
```

11.
```
#include<stdio.h>
 int main()
{
        char str[20] = "GeeksQuiz";
        printf ("%d", sizeof(str));
```

```
        return 0;
}
```
12.
```
#include<stdio.h>
int main()
{
        int a[2][3][2]={{{1,2},{9,8},{3,7}},{{2,2},{1,4},{5,4}}};
        printf("%ld\n",a[0]);
        printf("%ld\n",a[1]);
        printf("%ld\n",a[0][0]);
        printf("%ld\n",a[1][0]);
        printf("%d\n",a[0][0][0]);
        printf("%d\n",a[1][0][0]);
        printf("\n %d %d %d", a[1]-a[0],a[1][0]- a[0][0],a[1][0][0]-a[0][0][0]);
        return 0;
}
```
13.
```
#include<stdio.h>
int main()
{
        int arr[1]={10};
        printf("%d\n", 0[arr]);
        return 0;
}
```
14.
```
#include<stdio.h>
int main()
{
        int a[3][4] = {1, 2, 3, 4, 4, 3, 2, 1, 7, 8, 9, 0};
        printf("%u, %u\n", a+1, &a+1);
        return 0;
}
```
15.
```
#include<stdio.h>
int main()
{
        printf(5+"Good Morning\n");
        return 0;
}
```