Indian Institute of Information Technology Sri City, Chittoor

Examination: CP Lab-7
Duration: 3 Hrs.
Date: 07/02/2022
Maximum Marks: 20

INSTRUCTIONS:

1. Please carefully read all assignment problems and write the required programs in the C language.
2. All the PROBLEMS are COMPULSORY.
3. You should submit only a single C file containing all your answers. Make sure that during submission, no part of your code is commented.
4. **Name the file as follows: Roll No_Lab7.c**
5. **Do not upload zip files. Upload .c file and the screenshot of your terminal where you display the output. Upload the files in the submission link of CP Lab-7 in Google classroom.**
6. Don't share or copy the codes. If malpractice is found, you will be awarded zero.
7. If you do not follow the above-mentioned instructions, a strict penalty would be imposed.

ASSIGNMENTS:

1) Write a C program using functions for the problems given below.
   (a) Find first N (e.g., N =10) prime numbers greater than equal to X (e.g., X =5). Write functions **isPrime()** and **allPrime()** to check a prime number and to find N prime numbers respectively. **Print N prime numbers.** **(3 Marks)**
   (b) Using the N prime numbers obtained from (a), print all the pairs of prime numbers whose sum is divisible by x (e.g., x =2) and y (e.g., y=3). Write functions **pairPrimeDivisible()**, **sumTwoInteger()** and **isDivisible()** to find all pair of primes satisfying the above divisibility criteria, sum of two integers and check divisibility respectively. Call the function **pairPrimeDivisible()** from **allPrime().** **(4 Marks)**
   (c) Using the N prime numbers obtained from (a), print all the pairs of prime numbers whose sum minus 3 is also present in the array of N prime numbers. Write functions **pairPrimePresent()** and **isPresent()** to find all pair of primes satisfying the above criteria and to check whether the value is present or not in the array respectively. Use sumTwoInteger() function to find the sum of two prime numbers. Call the function **pairPrimePresent()** from **allPrime().** **(3 Marks)**

2) Write a C program using functions for the problems given below.

(a) Generate two Matrices: A of size M by M and B of size K by K by the following way. Here M and K are two odd numbers and M is greater than K (e.g., M=5 and K =3).

(i) Generate each element of matrix A by the expression (**rand()%(upper - lower + 1)) + lower**, where rand() is a random number generator (a library function in <stdlib.h>), lower = 0 and upper = 255. The expression will produce an integer value in between 0 and 255. **Print the matrix A.**

(ii) Generate each element of matrix B by the expression given below.

$$B(x, y) \;=\; \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

The value of σ is equal to 1.0, the value of π is equal to 3.14159 (M_PI in <math.h>) and exp() is the exponential function (exp() in <math.h>).

For matrix B in Q2, x and y value ranges from -n to n if (2n+1) by (2n+1) is the size of the matrix B. Note that, indices of B are 0 to 2n for row and column.

After that, divide each element of matrix B by the sum of all elements so that the sum of all elements becomes 1. **Print the matrix B.** **(3 Marks)**

For example, a matrix B of size 3 by 3 is

```
0.075114   0.123841   0.075114
0.123841   0.204180   0.123841
0.075114   0.123841   0.075114
```

(b) Write a function **fnMatrixOperation1(A, B)** which computes A*B as given below and stores the output in a matrix C of size (M-K+1) by (M-K+1). For example, if M=5, K=3 then the size of C is 3 by 3. **Print the matrix C.** **(4 Marks)**

For M = 3, K=2,

A * B =

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$$

where,

$$c_{11} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = a_{11}.b_{11} + a_{12}.b_{12} + a_{21}.b_{21} + a_{22}.b_{22}$$

$$c_{12} = \begin{pmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = a_{12}.b_{11} + a_{13}.b_{12} + a_{22}.b_{21} + a_{23}.b_{22}$$

$$c_{21} = \begin{pmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = a_{21}.b_{11} + a_{22}.b_{12} + a_{31}.b_{21} + a_{32}.b_{22}$$

$$c_{22} = \begin{pmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = a_{22}.b_{11} + a_{23}.b_{12} + a_{32}.b_{21} + a_{33}.b_{22}$$

(c) Add one extra row and column to C at the end with 0 entries to get a new matrix D of size (M-K+2) by (M-K+2). Then take 2 by 2 non overlapping submatrices in D and find the maximum value of the entries of the matrix to generate a new matrix E. Write a function **fnMatrixOperation2(C)** to find D and E. Call the function **fnMatrixOperation2(C)** from **fnMatrixOperation1(A, B). Print the matrices D and E.**

(3 Marks)

For example, if

$$C = \begin{pmatrix} 12 & 34 & 25 \\ 65 & 32 & 20 \\ 10 & 23 & 5 \end{pmatrix},$$

after adding one row and column,

$$D = \begin{pmatrix} 12 & 34 & 25 & 0 \\ 65 & 32 & 20 & 0 \\ 10 & 23 & 5 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

The final output,

$$E = \begin{pmatrix} \max\begin{pmatrix} 12 & 34 \\ 65 & 32 \end{pmatrix} & \max\begin{pmatrix} 25 & 0 \\ 20 & 0 \end{pmatrix} \\ \max\begin{pmatrix} 10 & 23 \\ 0 & 0 \end{pmatrix} & \max\begin{pmatrix} 5 & 0 \\ 0 & 0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 65 & 25 \\ 23 & 5 \end{pmatrix}$$

Note: You may take all the matrices of integer entries except the matrix B.