# PP Lab End-Sem ( CSE 3263)

**Date:** 11-06-2021
**Time**: 6 PM–7.30 PM (includes 15 mins for scan and upload time also)

**Instructions To students:**

- Only writeup and No execution
- Upload before close time without fail
- Upload the scanned legible handwritten answer in single pdf file
- File_name_format: RollNo_Name.pdf e.g : 23_Preethi.pdf

---

Write an efficient parallel CUDA program (host code + kernel code) which produces a two-dimensional character matrix ***RES*** of size ***NxN*** from a given input two-dimensional character matrix ***A*** of size ***NxN*** where ***N*** is an even number. Write a host code to read value of ***N*** and input matrix ***A*** with ***N*** strings where size of each string must be less than or equal to ***N*** including null character. Send the input matrix ***A*** to the kernel launched with (**2, 2) grid** and **2D block** and generate the output matrix ***RES*** as shown below in Sample I/O. Write the kernel code to incorporate the following conditions while producing every character of the output matrix ***RES*** in parallel. The host displays the input matrix ***A*** and the output matrix ***RES*** produced by the kernel.

1. All the border elements of the input ***A*** must be replaced with a special character '**!**' as shown in the output ***RES*** in **red** color.
2. All the unfilled positions of every row with row index as a prime number are replaced with a special character '*****' as shown in the output ***RES*** in **violet** color.
3. All the unfilled positions of every row with row index as a non- prime number are replaced with a special character '**#**' as shown in the output ***RES*** in **green** color.
4. All the vowels in the alphabetic positions of the input matrix ***A*** other than the positions mentioned in condition 1 should be toggled (i.e., uppercase to/from lowercase) as shown in the output ***RES*** in **blue** color.

Note: Kernel should accept only required arguments. Host code must allocate global memory for output matrix.

Sample Input : Enter *N* (an even number): 6
Enter the input character matrix ***A***:

**I**
**WriTE**
**pCaP**
**SEm**
**LAB**
**eXAMS**

Sample output ***RES***:

```
! ! ! ! ! !
! r I T e !
! C A P * !
! e m * * !
! a B # # !
! ! ! ! ! !
```