CS313: Intermediate Computer Programming

To Be Completed By: 23:30Hrs, March 25, 2021

Outcome

This is an individual assignment. After completing this exercise, you will be expected to have learnt and understood the following:

i. The use of the Unix/Linux programming environment. If your laptop already runs a Linux or a MacOS system you are good to go. If your laptop runs Windows, you have a choice of reconfiguring your system to run one of the Linux distros (Ubuntu 18.04/20.04 recommended):

- by dual booting.
- installing a virtualisation software, either VirtualBox or VMWare, and them installing a version of Linux as a virtual machine.
- installing a Windows-10 subsystem for Linux (WSL. You must be running the 64-bit Windows-10. The Ubuntu distro is recommended.
- installing the Cygwin in Windows-10.

ii. The use of one or more of the following editors – vi/vim/gvim, gedit sublimetext, atom or emacs;

iii. How to compile and run programs with the GCC/G++ compiler; a choice of one gcc/g++-7.5.X or gcc/g++-9.3.X for some simple C++ programs such as the one below of a "hello_world.cpp" code.

```
#include <iostream>
using namespace std ;
int main(int argc, char *argv[])
{
    if (argc <= 1) {
        cout << " Please specify at least one input value " << endl;
        cout << " Example: hellow_world John" << endl;
        cerr << " Program Terminating " << endl;
    }
    else {
        cout << "hello world " << argv[1] << endl;
        return 0;
    }
}
/*
    To compile
    % g++ -std=c++17 hello_world.cpp -o hello_world

    To run use:
```

```
                    % ./hello_world Name
                */
```

iv. How to use GIT and setup a project account on GitHub.

v. **Please include a documentation branch for your reports** in GitHub. Your entire project directory will be cloned for the marking.

Please keep in mind that this laboratory exercise and the subsequent ones are intended to prepare you for your projects. Please ensure that you save copies of your laboratory assignment on your **pen-drive** or **flash-drive** as backups.

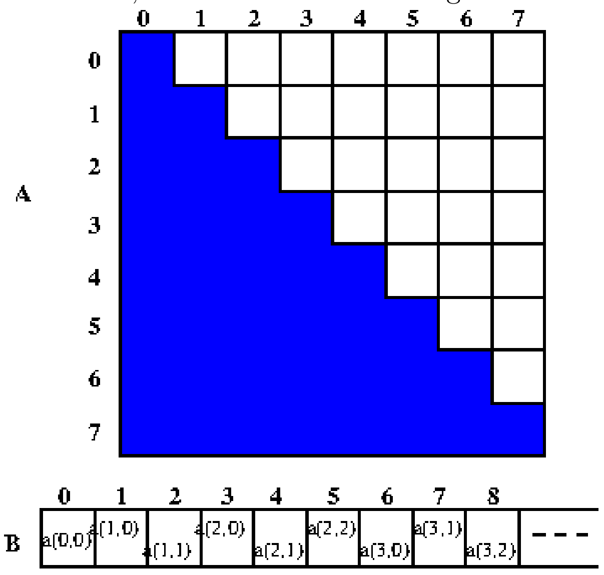
Work Schedule

Unix Tutorial for Beginners

If you are not already familiar with Unix then consult any Unix tutorial site and drill yourself on how to use this environment. Familiarise yourself with basic commands such as: ”ls, cd, mkdir, pwd, cat, less, etc.”

Problem Description: Mapping a lower-triangular array onto consecutive linear storage

You are required to write a main procudure and 2-functions **LiearIndx(),** and **InverseIdx()** all in C++. The main generates and stores a lower triangular array $A[N][N]$ for a given N. The elements of the lower triangular array are to be stored in consecutive locations of a 1-dimensional array $B[]$ definitely of size much less than N^2, as illustrated in the diagram



The tasks are:

A) Write a function LinearIndx(..., i, j...) that takes the integer indices i and j of an aarray element A[i][j] and returns the index I of the entry in the 1-dimensional array, B[] where the element is stored. i.e.,

$$B[I] = A[i][j]$$

B) Write a function InverseIdx(...,I,...) that takes an index I of an element B[I] and computes the reverse indices i, j. of the corresponding element in A[][].

You are then to write a main program, that dynamically generates in turn, 3 lower triangular arrays, A[8][8], A[32][32] and A[128][128] where only the loweer-trangular elements have values. The others are zeroes or undefined. For each size of the array the main program should fill the lower-trangular elements with random integer numbers X between 1 and 1000, i.e., $1 \leq X \leq 1000$

The elements of the array A[][] are then assigned into the correct position in B[]. Declare a third array C[N][N] and for each the element in B[I], make a copy of it into the correct corresponding location C[i][j].

The Deliverable

- Submit your computer codes for marking to your initialised repository on GitHub.
- Provide a short description of your solution to the linearization function of the lower-triangular array. Give the formulae F() used to compute $I = F(i, j)$ and describe inverse procedure to compute $(i, j) = F^{-1}(I)$
- For each value of N
  - print the first 20 values of the lower trinagular elements of A[][] and their respective indices in row-major order;
  - print the first 20 elements of B[] and their respective indices;
  - print the first 20 elements of C[][] and their respective indices;
- Write a short set of instructions on how to access your GitHub repository and submit this to CANVAS.