**CECS 424**
**Assignment 8**
**Total: 50 Points**

---

General Instruction

- If your programs could not be executed on a command line environment, **zero** grade will be given for the programs.
- Submit uncompressed file(s) via BeachBoard (Not email or in class).

---

1. (20 points) Write a program that does matrix multiplication in `C++` without using any external library.

   i. The file names of the source codes should be `Assn8.cpp`.

   ii. The program should read the command-line arguments passed to the program. (`argv[]`).

   ```
   argv[1]: number of rows in the matrix 1 (r1)
   argv[2]: number of columns in the matrix 1 (c1)
   argv[3]: number of rows in the matrix 2 (r2)
   argv[4]: number of colums in the matrix 2 (c2)
   ```

   iii. The program should allocate a **heap memory** space dynamically for the matrices in the row major order.

   iv. The program should **not** use a `VLAs (Variable-Length Array)` even it is supported by the `GNU C Compiler`.

   v. The program should **not** use the bracket operator(`[]`) to access the each element of the matrices, please use pointers, pointer arithmetics, and dereference operators. (`arr[i][j] -> *(*(arr + i) + j)`)

   vi. The program should assign the numbers $\{1, 2, 3, \cdots, r_1 \times c_1\}$ to the **first** matrix in **ascending** order. For $3 \times 3$ example,

   ```
   1 2 3
   4 5 6
   7 8 9
   ```

   vii. The program should assign the numbers $\{1, 2, 3, \cdots, r_2 \times c_2\}$ to the **second** matrix in **descending** order. For $3 \times 3$ example,

   ```
   9 8 7
   6 5 4
   3 2 1
   ```

   viii. Execution command and the expected output would be:

```
g++ Assn8.cpp -o Assn8; ./Assn8 3 3 3 3
Matrix 1
1 2 3
4 5 6
7 8 9
Matrix 2
9 8 7
6 5 4
3 2 1
Matrx 1 * Matrix 2
30 24 18
84 69 54
138 114 90
```

ix. Submit your source code, `Assn8.cpp`.

2. Write down your answers to the questions and submit a `PDF` file.
   (a) (10 points) Translate the following expression into postfix and prefix notation.

$$(b * b - 4 * a + c)/(2 * a)$$

   (b) (10 points) Consider the following program in `C++`. What will be the final values of `fp_count` and `int_count`. Run the program in your system and explain your answer.

```
int fp_count = 0, int_count = 0;
for (float i = 0; i < 1; i += 0.01) {
   fp_count++;
}
for (int i = 0; i < 100; i += 1) {
   int_count++;
}
```

3. (10 points) Find the `Assn8.hs` and complete the definition of the tail recursive `tailFac` and the `tailFib` functions. Submit your `Assn8.hs`.