EE231002 Introduction to Programming

Lab07. Latin Squares

Due: Nov. 27, 2021

A Latin Square is an $N \times N$ matrix with N different symbols such that each row and column has the N different symbols occurring exactly once and without repetitions. For example, the followings are 3×3 Latin Squares.

$$\begin{bmatrix} A & B & C \\ B & C & A \\ C & A & B \end{bmatrix} \qquad \begin{bmatrix} A & B & C \\ C & A & B \\ B & C & A \end{bmatrix} \qquad \begin{bmatrix} A & C & B \\ B & A & C \\ C & B & A \end{bmatrix}$$

where we have used the first 3 alphabet letters for the symbols. In this assignment, please write a C program to perform **exhaustive** search for **all** possible Latin Squares given N, which is defined by macro as below, and the symbols are consist of the first N capital alphabets.

```
#if !defined(N)
#define N 3
#endif
```

Defining macro as shown above, it is possible to change the value of N while compiling. For example, if we want to change N from 3 to 4 we just need to do the following.

```
$ gcc -DN=4 lab07.c
```

Then during preprocessing phase of compilation, the value of the macro is replaced by 4. You will need to make sure your program can handle general cases with $N \geq 3$. Example of program outputs are (N=3):

```
Latin Square 1:
  A B C
  B C A
  C A B
Latin Square 2:
  A B C
  C A B
  B C A
Latin Square 3:
  A C B
  B A C
  C B A
. . .
. . .
```

1

```
Latin Square 12:
  C B A
  B A C
  A C B
Totoal number of Latin Squares found is 12
```

Notes.

1. Create a directory **lab07** and use it as the working directory.

2. Name your program source file as **lab07.c**.

3. The first few lines of your program should be comments as the following.

   ```
   // EE231002 Lab07. Latin Squares
   // ID, Name
   // Date
   ```

4. It is recommended to define the two-dimensional matrix and a integer variable as global variables such that they can be accessed by all functions.

5. **Functions** have been discussed in the class, and you are free to define and use necessary functions, including recursive functions, in your codes.

6. After finishing editing your source file, you can execute the following command to compile it,

   ```
   $ gcc lab07.c
   ```

   If no compilation errors, the executable file, **a.out**, should be generated, and you can execute it by typing

   ```
   $ ./a.out
   ```

7. After you finish verifying your program, you can submit your source code by

   ```
   $ ~ee2310/bin/submit lab07 lab07.c
   ```

   If you see a "submitted successfully" message, then you are done. In case you want to check which file and at what time you submitted your labs, you can type in the following command:

   ```
   $ ~ee2310/bin/subrec lab07
   ```

   It will show the last few submission records.