Programming I – Project 1
Due: November 22 by 17:00 in edubiz and hard copy to me.

Project Description:
You will work with a table and apply some operations on this table. The following table that contains integers between 0 and 9 inclusive are generated by a random number generator. Study the rand function that generates random numbers. The function is declared in the stdlib.h file. Implement the following list in order. Do not start a new step before finishing all before that. **In each step of your implementation you need to use loops.**
NOTE: Turn in this page as well with your deliverables at the end of this document. **Mark all the steps (1 through 9) on this page that you successfully completed by circling them.** The total points are given at the end of items so that you will know your maximum grade when you turn in the project. Failure to answer any of the questions I may ask will result in a failure in the project. By turning in what you finished you claim that this is your work so that you should be able to answer any questions. Remember that this project should be done individually and searching/reading/studying the internet is encouraged and you can even discuss with me your implementation to get my opinion on it.

1. Generate a table of size 8 rows and 8 columns with random numbers between 0 and 9 inclusive. See Figure 1 for a table that is generated randomly. You will generate the same table even though the table is generated by the rand function. (10 points).

2. Find the row sums in the table. Check Figure 2 for a sample output. (15 points)

3. Find the column sums in the table. Check Figure 3 for a sample output. (20 points)

4. Create a new table where new table's row r and col c will be assigned 9 if the original randomly generated table's row r and col c contains a number greater than 4, 0 (zero) otherwise. Print the new table to the terminal. Check Figure 4 for a sample output of the original table. (25 points)

5. Find the count of every digit in the table and print them to the terminal. Check Figure 5 for a sample output. (30 points)

6. Do (5) but this time show the count of every digit in a tabular form with the format explained below. Note that the first column in Figure 6 is the count of a digit, the last row is the digit itself and the 'x's are showing the count of that digit. For example, the number of 'x's for digit 0 is 6 since there are 6 'x's vertically on the column for digit 0 which is shown with a red box that encloses those x's. Compare the counts in (5) and Figure 6 to see they are the same numbers represented differently, that is, with numerals in (5) and 'x's with Figure 6. Check Figure 6 for a sample output. Note that the tabular form that shows counts of each digit in the table should not be hard coded, that is, if the table changes (for example your random number generator generates different digits then your program should print the right data statistics). (50 points)

7. Similar to (6), you need to show the counts of two consecutive digits, namely, count of 0 and 1 digits in the table, count of digits 2 and 3 in the table and so on. Check Figure 7 for a sample output. The bottom row in Figure 7 shows the larger of the two digits you count. 1 is shown for the sum of digits 0 and 1, 3 for the sum of digits 2 and 3 and so forth. (60 points)

8. You will find the maximum of every 2x2 grid in the 8x8 table. Then you will print the maximum of every 2x2 grid to the terminal in the order they are in the table. The order of these 2x2 grids are from top left corner to bottom right shown in Figure 9. These 2x2 grids should be handled within a loop(s), that is, visiting each 2x2 grid should be done using a loop(s). Once you are on a grid you can find the largest of the 4 numbers (there are 4 numbers in a 2x2 grid) by any way/method you want (no need for a loop in this case). The output is shown in Figure 9. (80 points)

9. For this last item, you will read a 2x2 grid from the keyboard. Then you will go through all 16 2x2 grids in the original table and find the “the most similar” grid to the 2x2 grid read from the keyboard. Two most similar 2x2 grids are computed by finding the difference (disregarding the sign, that is, absolute value) between all four corresponding pairs of numbers in each of the two grids. At the end of finding the most similar 2x2 grid to the one entered from the keyboard, print them to the terminal. Figure 10 shows a sample output for the original table generated randomly. Assume that the first grid printed is the one read from the keyboard. (100 points)

```
~/Documents$ ./a.out
3 6 7 5 3 5 6 2
9 1 2 7 0 9 3 6
0 6 2 6 1 8 7 9
2 0 2 3 7 5 9 2
2 8 9 7 3 6 1 2
9 3 1 9 4 7 8 4
5 0 3 6 1 0 6 3
2 0 6 1 5 5 4 7
```

Figure 1. Randomly generated table (printed to the terminal). a.out is the name of my executable program that prints the table above after randomly generating each cell in the table.

```
row sums:
row[0] sum=37
row[1] sum=37
row[2] sum=39
row[3] sum=30
row[4] sum=38
row[5] sum=45
row[6] sum=24
row[7] sum=30
```

Figure 2. Summation of each row. row[0] (first row's summation of all digits) sum is 37, etc…

```
col sums:
col[0] sum=32
col[1] sum=24
col[2] sum=32
col[3] sum=44
col[4] sum=24
col[5] sum=45
col[6] sum=44
col[7] sum=35
```

Figure 3. Summation of each column. col[0] (first column's summation of all digits) sum is 32, etc…

```
0 9 9 9 0 9 9 0
9 0 0 9 0 9 0 9
0 9 0 9 0 9 9 9
0 0 0 0 9 9 9 0
0 9 9 9 0 9 0 0
9 0 0 9 0 9 9 0
9 0 0 9 0 0 9 0
0 0 9 0 9 9 0 9
```

Figure 4. The new table is printed to the terminal. The new table now contains only 0's or 9's. Any number in the original table greater than 4 is now 9 and 0 otherwise in the new table.

```
count of 0 = 6
count of 1 = 6
count of 2 = 9
count of 3 = 8
count of 4 = 3
count of 5 = 6
count of 6 = 9
count of 7 = 7
count of 8 = 3
count of 9 = 7
```

Figure 5. Count of each digit in the table. If you look at the table, there are 6 zeros, ones and fives, there are 9 twos and sixes, etc…

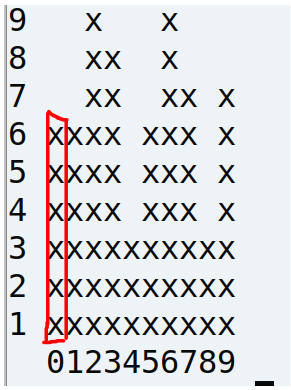

Figure 6. Count of each digit in the table shown in a tabular form.

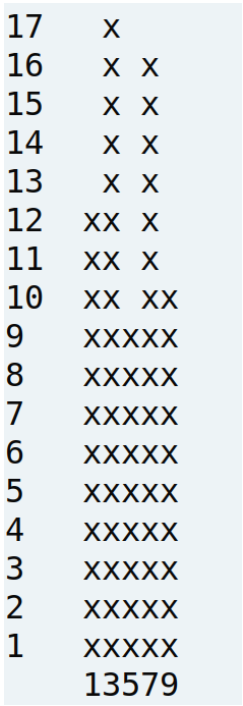

Figure 7. Count of each “pair of digits” in the original table

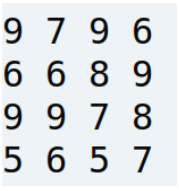

Figure 8. Max of every 2x2 portion of the table randomly generated shown in Figure 1.

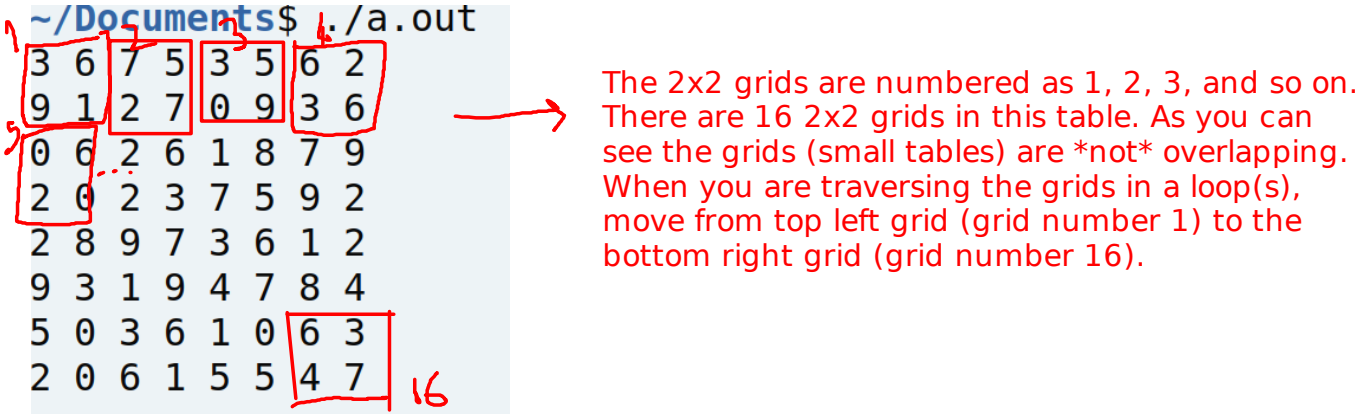

Figure 9. 2x2 grid are marked with red boxes. The flow from top left to bottom right.

```
Please enter a 2x2 grid to search closest in table:
2 5
2 2
md_grid:
2 6
2 3
```

Figure 10. A 2x2 grid entered and closest 2x2 grid from table.

Deliverables: Please turn in your soft copy of your .c file in edubiz.

On the due date (November 22 Wednesday) please bring me a hard copy of the following during my office hour. If you finish the project earlier, you are more than welcome to bring it to me any time.
Turn in a hard copy of your source code (.c file) with the cover page (project description with all items 1 through 9) and circle those steps that you finished. In your source code mark your implementation with the step numbers. Also after each step (1 through 9) in your source code, after your code, put your output as comments.