Lab 5: Reviewing Exam 1

1. What does the following program print? Enter and run the program and write down the result.

```
#include <stdio.h>

int main(void) {

   int x = 2;

   x *= 3 + 2;
   printf("%d\n", x);

   x = -3 + 4 * 5 – 6;
   printf("%d\n", x);

   x = 3 + 4 % 5 – 6;
   printf("%d\n", x);

   x = -3 * 4 % -6 / 5;
   printf("%d\n", x);

   x = (7 + 6) % 5 / 2;
   printf("%d\n", x);

   return 0;

} // main()
```

2. What does the following program print? Enter and run the program and write down the result.

```
#include <stdio.h>

int main(void) {

   int x = y = z = 1;

   x += y += z;
   printf("%d\n",  (x < y ? y : x) );

   printf("%d\n", (x < y ? x++ : y++) );
   printf("%d\n", x);
   printf("%d\n", y);

   x = 3; y = z = 4;
   printf("%d\n", (z >= y >= x) ? 1 : 0) );

   return 0;

} // main()
```

3. Write a **complete C** program, called **letters**, that inputs a stream of text (one character at a time) from the standard input and counts the frequency of occurrence of the letters **A** (or **a**) through **Z** (or **z**).

4. Write a **complete C program** that simulates the rolling of a die (outcomes **[1..6]**) **n** times and records the number of occurrences of each outcome. From this data, it should compute and print the empirical probability of getting an outcome of **3**. For example, if **n = 10** and the outcomes are:

```
5  1  4  5  3  3  2  6  1  4
```

The empirical probability of getting a 3, based on the above is **0.20** (up to two decimal places).

Write the program for **n = 10** million.

5. Complete the definition of the **C function** below and test it in a program. It determines the largest elements in each row and places them in an array. For example,

```
int data[4][4] = { {1, 5, 3, 2},
                   {4, 6, 1, 3},
                   {9, 2, 6, 7},
                   {5, 8, 3, 4} };

int rMax[4];
```

After the call:

```
maxRow(4, data, rMax);
```

We will have:

```
rMax = [5, 6, 9, 8]
```

Here is the function definition you have to complete:

```
void maxRow(int n, int A[n][n], int rm[n]) {

   // determines the max element in each row of A[][] and places it in rm[].
```