CS 100 Exam Three – Coding – Fall 2017

You cannot use the Internet while coding the two problems below.
You can log into the cs-intro server to test your programs.
When you have finished coding, submit your exam via Blackboard

Create a directory called `exam3` using `mkdir exam3` and move into that directory with `cd exam3`

Complete the two programs shown below. When you are finished, submit your exam to Blackboard.

- First, on your local machine, compress your `exam3` directory into a single (compressed) file.
- Second, once you have a compressed file named `exam3.zip`, submit that file to Blackboard.

---

1. Name this program **one.c** – This program counts the number of occurrences of a given character in the command line arguments, stopping when the user enters **<control-d>**. When counting, do not count any characters in `argv[0]` (the name of the executable, which is normally either `a.out` or `a.exe`). Just count occurrences in the other arguments on the command line. A sample execution of the program is shown below:

```
./a.out Crimson Tide University of Alabama 1831
Enter a character (or <control-d> to exit): e
        e occurs 2 times
Enter a character (or <control-d> to exit): a
        a occurs 3 times
Enter a character (or <control-d> to exit): i
        i occurs 4 times
Enter a character (or <control-d> to exit): 1
        1 occurs 2 times
Enter a character (or <control-d> to exit):
                                   <control-d>
```

2. Name this program **two.c** – This program reads the name of a square two-dimensional array, and the size of this array, from the command line. It reads the array into memory and then calls a function that computes the sum of all values on the borders (edges) of the array. In the two examples below, the border elements are the elements shown in red.

```
 1  2  3  4
 5  6  7  8
 9 10 11 12
13 14 15 16
```

Given the file `DATA1` shown at the left, running the program `./a.out DATA1 4` would result in the answer `102`

Given the file `DATA2` shown at the right, running the program `./a.out DATA2 3` gives the answer `40`

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

The code below can be downloaded from Blackboard. You must write the function `sumBorders` (everything else already exists). **Hint**: make sure you handle the special case of a 1x1 matrix.

```
#include <stdio.h>
#include <stdlib.h>

int sumBorders(int **, int);

int main(int argc, char *argv[]) {
    FILE *fp = fopen(argv[1], "r");
    int size = atoi(argv[2]);
    // allocate the matrix
    int **matrix = malloc(sizeof(int *) * size);
    for (int a=0; a<size; a++)
        matrix[a] = malloc(sizeof(int) * size);
    // read the matrix
    for (int a=0; a<size; a++)
        for (int b=0; b<size; b++)
            fscanf(fp, "%d", &matrix[a][b]);
    // compute the answer
    int answer = sumBorders(matrix, size);
    printf("The answer is %d\n", answer);
    return 0;
}
```