CS 100 Lab Ten – Spring 2019

Create a directory called **lab10** on your machine using **mkdir lab10**. Please note this lab is about recursive functions, and no credit will be given if a function is completed iteratively instead of recursively.

1. Name this program **one.c** – This program takes a non-negative integer as its command-line argument and counts the number of ones (1s) in the binary representation of the integer. You are asked to complete a recursive function named **ones** to count the number of ones in the binary representation of its parameter **num**. For example, binary representations of 5, 15 and 1020 are 101, 1111, and 1111111100 respectively. When the program is run with these integers, the program will produce the printouts are shown below at the right.

```
#include <stdio.h>
#include <stdlib.h>
int ones(int num) {
    // add your recursive code here
}
int main(int argc, char *argv[]) {
    int val=atoi(argv[1]);
    printf("%d in binary contains %d ones\n",
            val, ones(val));
    return 0;
}
```

```
./a.out 5
5 in binary contains 2 ones

./a.out 15
15 in binary contains 4 ones

./a.out 1020
1020 in binary contains 8 ones
```

2. Name this program **two.c** – This program finds the largest character (the character with the largest ASCII value) of all the command line arguments, excluding the command name. You are asked to write two recursive functions, **maxCharOfString** and **maxCharOfStringArray**. **maxCharOfString** is to find the largest character out of a single string and then it is used to implement **maxCharOfStringArray** that is to find the largest character out of an array of strings. Two sample executions of the program are shown below.

```
#include <stdio.h>
// Find the largest character of a non-empty string
char maxCharOfString(char *str) {
    // add your recursive code here
}
// Find the largest character of an array of strings
char maxCharOfStringArray(int num, char *array[]) {
    // add your recursive code here
}
int main(int argc, char *argv[]) {
    if (argc==1) {
        printf("%s requires at least one argument\n", argv[0]);
        return 1;
    }
    printf("The largest character of all arguments is %c\n",
            maxCharOfStringArray(argc-1, argv+1));
    return 0;
}
```

0 1 2 3 4

./a.out UNIVERSITY OF ALABAMA TUSCALOOSA

The largest character of all arguments is Y

./a.out ROLL TIDE

The largest character of all arguments is T

Submit your lab

First, on your machine, compress your **lab10** directory into a single (compressed) file, i.e. **lab10.zip**.

Second, once you have a compressed file named **lab10.zip**, submit that file to Blackboard.