C Arrays and I/O
Lab E

TA teaching instructions can be found at the end of this document.

The labs, for this course, are designed to be completed on your own at home or in the 3rd floor Trottier labs. These labs are not graded. You do not hand in these labs. If you prefer to work on a lab with your TA tutorial group, then check the schedule for your TA's tutorial session. You will find this schedule in our MyCourses page under Content/Course Information/TA Information. Since the university has limited lab space, your TA might ask you to bring your laptop and work in a classroom instead of a lab.

This lab is about programming C arrays and basic I/O commands.

Some labs will have a question zero. These questions will not be covered by the TA during the tutorial. It is extra content meant for you to do on your own.

QUESTION ZERO: Optional problem

Review the class notes about C arrays and the `stdio.h` functions.

QUESTION ONE: Arrays and I/O

Given the following program:

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

Int main(int argc, char *argv[]) {
    char name[30], multipleNames[1000];
    int numberOfTimes;

    if (argc == 1) numberOfTimes = atoi(argv[1]);
    else numberOfTimes = 2;

    Printf(“Enter a name: “);
    scanf(“%s”, name);

    // Copy the word in the array name into the array
    // multipleNames for numberOfTimes

    Printf(“%s\n”, multipleNames);

}
```

Complete the above program.

For example: assume the user inputs `Bob` into the array `name[]`. Assume the user inputs 3 from the command line. The array `multipleNames[]` must contain `BobBobBob`.

You must write this algorithm without using additional library functions from string.h and stdio.h (for example, you cannot use the sprint() function, you cannot use the strcpy() function).

The source file name must be `names.c`. The compiled program must be `names`.

Hint: Look at this code –

```
char array[50];
char s[] = {‘a’,’b’,’c’,’\0’};

for(i=0; i<strlen(s); i++) array[i] = s[i];
for(j=i, i=0; i<strlen(s); i++, j++) array[j] = s[i];
array[j] = ‘\0’;
```

What does the above code put into array[] ?

What would the algorithm look like to compute the length of a string? **Hint**: how could a for-loop be used to count the number of characters in a null terminating string?

QUESTION TWO: The command getc()

Given the completed program from Question One and the Hint:

Replace the line of code that uses `scanf()` to input a word into the array `name[]`, with an algorithm that uses `getc(stdin)` to input a word into the array `name[]`.

How can you read a word of characters using `getc(stdin)`?

Hint: Look at this code, what does this code do?

```
char array[50];
int n=-1;

do { n++; array[n] = getc(stdin); } while(array[n] != ‘\n’);
```

You have completed your lab.

TA Teaching Instructions

The lab time is divided into two 30-minute periods. The first period is a lecture. The second period is the lab. The lab period is conducted in a TA directed setting where students do the above lab together with the TA. The first period is conducted as a lecture and covers the following material:

- Review how to construct and use an array
- Help them do this lab
- Then do a simple matrix multiply example with 2D arrays