C Control Statements and Arrays
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 on the 3rd floor Trottier labs. These labs are not graded. You do not hand in these labs. If you prefer to work in a lab with the TA tutorial group, then check the schedule for the TA's tutorial session. You will find this schedule in our MyCourses page under Content/Course Information/Prof & TA Coordinates. 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

Your solution program must fit completely within the main() function. **You cannot use helper functions**.

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 == 2) 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 library functions from string.h and stdio.h (for example, you cannot use the sprintf() function, you cannot use the strcpy() or strcat() functions). The only functions from stdio.h you are permitted to use are the ones already in the example code above. **You must manipulate the contents of the string using only loops, if-statements, pointers, and array index numbers.**

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 an algorithm look like to compute the length of a string using the above technique of manipulating strings? **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, update the main() function to do the following **without helper functions**:

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');
```

The source file name must be `names2.c`. The compiled program must be **`names2`**.

You have completed the lab.

TA Teaching Instructions

Divide the lab time into two 30-minute periods.

In the first 30 minutes, review the following:

- Write and explain a code snip that searches an array of characters terminating with a null for a single character. Show how a loop can find the first occurrence of the letter. Then modify this code so that it can count the number of times the character appears in the string. Do not use string.h. Solve this with only scanf, getc, and C control structures.
- Write and explain a code snip that performs a simple matrix multiplication with a 1D array and a single number. Then show this with a number and a 2D array. The goal is to demonstrate how one iterates through arrays to compute a solution matrix.

In the second 30 minutes:

- Help the students write the code for this lab as an activity. Ask the students for suggestions on how to solve this problem. Discuss their suggestions and write the solution code with them.