Introduction to C
Lab 5

The labs, for this course, are designed to be completed on your own at home of 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 in a supervised setting, check the TA Information schedule for the Lab TA period(s). You will find this schedule in our MyCourses page under Content/Course Information. The supervised labs are not teaching environments. The Lab TA will simply be present to answer questions and provide support.

This lab is about compiling and running your first C program.

PART ONE: Compiling a C program

Try this experiment:

(A) Login to mimi.cs.mcgill.ca
(B) In your home directory `mkdir` a directory called lab5
(C) Then `cd` into that directory
(D) Using `pico` or `vim` create a file called hello.c
(E) Type in the following program:

```
#include<stdio.h>
int main() {
  printf(“Hello World!\n”);
  return 0;
}
```

(F) Save the above file
(G) At the command line type: `gcc hello.c`
(H) There should be no error messages, if so go to step I else go to step E.
(I) Run the program by typing at the command line prompt: `./a.out`
(J) You should see the words `Hello World!` display on your screen. If so, congratulations you made your first program. If not go back to E and try again.
(K) If you do not like the a.out filename, you can recompile with this: `gcc -o hello hello.c` This will create an executable program with the name hello from the file hello.c. You can use any name you like; this lab uses hello. To run the program type: `./hello`

PART TWO: Arrays and Input

Look at the following code:

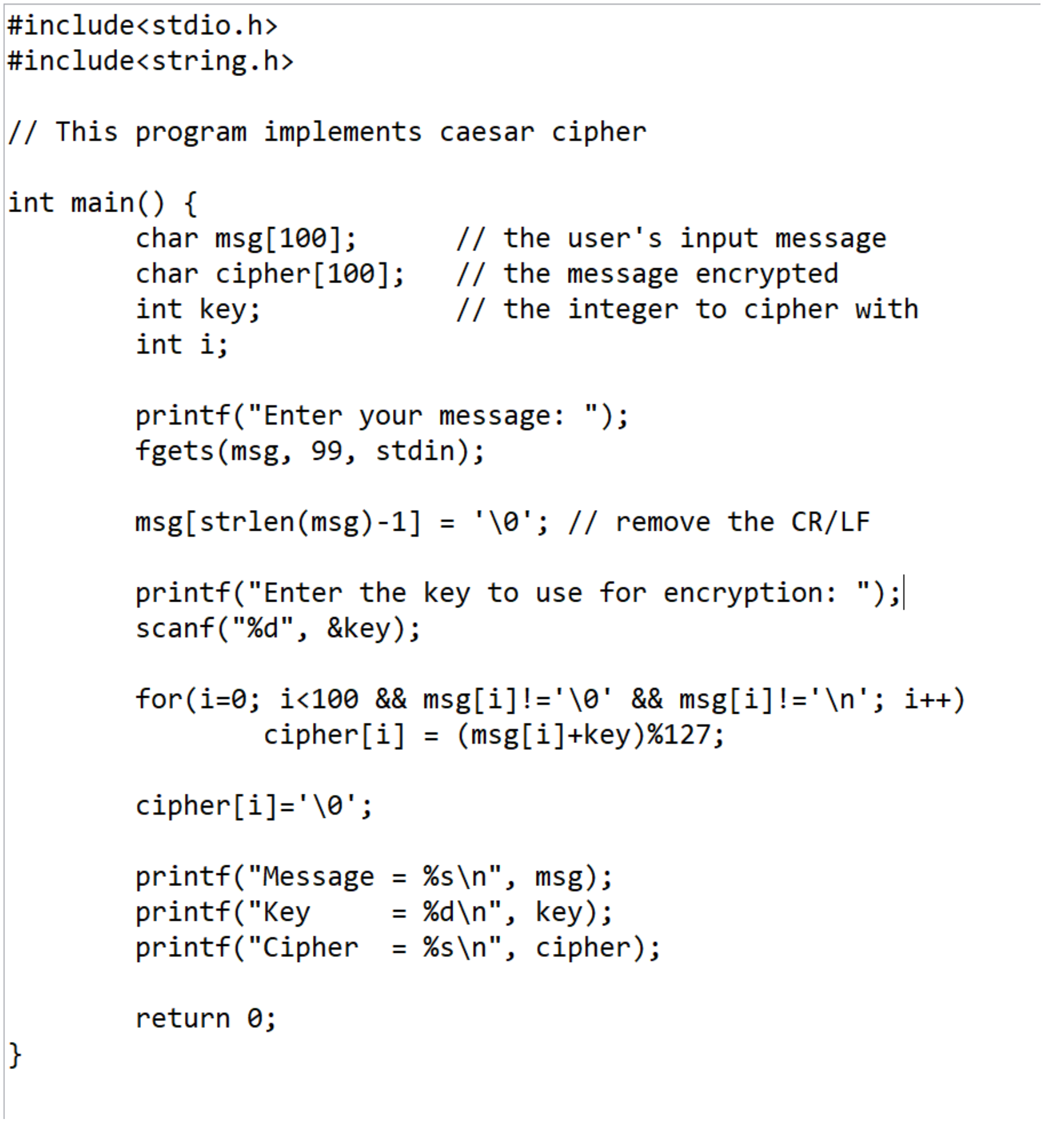
```
#include<stdio.h>
#include<ctype.h>

int main() {
        int array[] = {5,10,15,20};
        int arraylength = 4;
        int choice, i;
        int done = 0; // where 0 means FALSE
        char c;
        char garbage;

        printf("Check to see if your number is in the array!\n");

        while(!done) {
                printf("Input your number: ");
                scanf("%d", &choice);
                garbage = getc(stdin);

                for(i=0; i<arraylength; i++)
                        if (array[i] == choice) break;

                if (i==arraylength)
                        printf("SORRY! Your number is not in the array.\n");
                else
                        printf("YES!! You found a number in the array.\n");

                printf("Look for another number? (Y/N): ");
                scanf("%c", &c);
                if (toupper(c) == 'N') done = 1;
        }

        printf("Bye\n");

        return 0;
}
```

Answer the following questions:

(A) From a visual inspection, what does the program do?
(B) From a visual inspection, what does the garbage variable do?
(C) In your lab5 directory, type in and compile the above program. Call it find.c .
(D) Get the program to run correctly
(E) What does the toupper() function do?

PART THREE: Manipulating characters and arrays

Look at the following code:



```c
#include<stdio.h>
#include<string.h>

// This program implements caesar cipher

int main() {
        char msg[100];      // the user's input message
        char cipher[100];   // the message encrypted
        int key;            // the integer to cipher with
        int i;

        printf("Enter your message: ");
        fgets(msg, 99, stdin);

        msg[strlen(msg)-1] = '\0'; // remove the CR/LF

        printf("Enter the key to use for encryption: ");
        scanf("%d", &key);

        for(i=0; i<100 && msg[i]!='\0' && msg[i]!='\n'; i++)
                cipher[i] = (msg[i]+key)%127;

        cipher[i]='\0';

        printf("Message = %s\n", msg);
        printf("Key     = %d\n", key);
        printf("Cipher  = %s\n", cipher);

        return 0;
}
```

Answer the following questions:

(A) From a visual inspection, what does the program do?

(B) From a visual inspection, what does `msg[strlen(msg)-1]` do?

(C) In your lab5 directory, type in and compile the above program. Call it caesar.c .

(D) Get the program to run correctly

(E) Add to this program; create a new array called char msg2[100]. Using the cipher array, decrypt the cipher into the array msg2. Then print out msg2. If you did it correctly, then the output from msg2 should be the same as msg.