Assignment 2

In this assignment you will use an editor and the gcc compiler to create some programs. For a maximum B+ grade, select any two of the choices below. You can get an A grade by either completing three programs with excellent work or by completing the fourth program successfully.

To turn in your homework, if you are on the cslinux computer, copy the source files to your local system using `scp`. Then zip them and submit to canvas.

Here are the programs to choose from:

1. Write a function to recognize digits.

to begin, copy the file `find_letters.c` to a new file `find_letters_digits.c` as follows:

```
>cp find_letters.c find_letters_digits.c
```

Take a look at the functions `isupperalphabetic` and `isloweralphabetic` in the source code file `find_letters_numbers.c`. Use the structure of those functions and the ASCII code table in Lecture 3 to create a function called `isnumber` that can tell if a character is a digit. This is just like testing for an upper or lower case letter. Add this function to `find_letters.c` and modify the `main` function as follows:

Modify the string `testStr` to contain some numbers. Modify `main` to include an `if` statement that checks `isnumber` and prints something to indicate when a digit is found. Make this like the printed results for finding letters.

The output should be similar to this:

```
string = 42 is the answer to life, the universe, and everything.
55
DIGIT: 4
DIGIT: 2
LETTER: i
LETTER: s
LETTER: t
LETTER: h
LETTER: e
```

2. Print an ASCII chart of Decimal, Octal, Hexadecimal, and ASCII Code

Begin by copying the file `find_letters.c` to make a file called `find_letters_ascii_codes.c` as follows:

```
cp>cp find_letters.c find_letters_ascii_codes.c
```

look at the file `integer_bases.c` in the `Hello` directory on github. This shows the codes for printing a number in different bases. In this exercise, you will modify the `main` function in this area:

```
  for (i = 0; i < len; i++){
    char c = testStr[i];
    if (isalphabetic(c)) {
      printf("LETTER: %c\n", c);
    }
  }
```

Change the line with `printf("... %c", c);` to also print the character c in decimal, octal, and hexadecimal bases.

The output should be something like this:

```
string = When shall we three meet again In thunder, lightning, or in rain?
65
LETTER: 87, 0127, 0x57, W
LETTER: 104, 0150, 0x68, h
LETTER: 101, 0145, 0x65, e
LETTER: 110, 0156, 0x6e, n
LETTER: 115, 0163, 0x73, s
... etc.
```

3. Write a function to recognize whitespace

This is really similar to problem 1. Copy `find_letters.c` (or `find_letters_numbers.c` if you want to include this with assignment 1) to a new file called `find_letters_whitespace.c`. Modify the file to include a function called `iswhitespace` that takes a `char` parameter and returns FALSE if the input is not a whitespace character and TRUE if it is. You can see which ASCII codes represent whitespace by look at the table on slide 5 of lecture 3.

Change the `main` function to show that you found whitespace (similar to problem 1) by calling `iswhitespace` in an if statement. If `iswhitespace` is true, print “WHITE” on a separate line between letters on the output. Your output should look like this:

```
string = When shall we three meet again In thunder, lightning, or in rain?
65
LETTER: W
LETTER: h
LETTER: e
LETTER: n
WHITE:
LETTER: s
LETTER: h
LETTER: a
LETTER: l
LETTER: l
WHITE:
LETTER: w
LETTER: e
WHITE:
.... etc.
```

4. Count letters and digits in a string and print the totals

For this program, start by copying `count_letters.c`. This program was discussed in Lecture 3. copy the program to create a file called `count_letters_digits.c`.

You will modify the file as follows.

First create a function called `isnumber` to recognize digits as in program 1 above.

Next modify main to contain a new array variable called `digits` that stores 10 `int`s. Add a `for` loop to initialize each entry in the array to zero.

Next modify `main` to have the following string:

```
  char testStr[] = "Accepting the absurdity of everything \
around us is one step, a necessary experience: it should not \
become a dead end. It arouses a revolt. 12345 + 678910 = 691255";
```

Next, further modify main to do for digits what it does for characters. every time a digit is seen, increment the count for that digit in the array.

Finally, print the digit and the number of that digit found in the string, exactly as the program does for characters. the output should look something like this:

```
string = Accepting the absurdity of everything around us is one step, a necessary experience: it should not become a dead end. It arouses a revolt. 12345 + 678910 = 691255
162
character: a appears 9 times.
character: b appears 2 times.
character: c appears 5 times.
character: d appears 6 times.
character: e appears 18 times.
character: f appears 1 times.
...
character: x appears 1 times.
character: y appears 3 times.
character: z appears 0 times.
digit: 0 appears 1 times.
digit: 1 appears 3 times.
digit: 2 appears 2 times.
digit: 3 appears 1 times.
digit: 4 appears 1 times.
...
```