Coding Assignment 4

Step 1 – Alter your library files - `MyLib.c` and `MyLib.h` file.

1.   `ConvertDecimalToBinary()` should only take one parameter now –the decimal number to be converted

2.   `ConvertDecimalToBinary()` should convert the decimal number to binary and print it (move the code from `PrintBinary()` into `ConvertDecimalToBinary()`).

3.   Eliminate `PrintBinary()` now that `ConvertBinaryToDecimal()` prints the binary number.

Step 2 – Create function `GuessALetter()` in your `Code4.c` file

Create a function called `GuessALetter()`. It takes two parameters, a character array named `Phrase` and a character array named `PhraseCopy`. It has a return value of type `int`.

Create 4 variables

      a character variable named `Guess`

      a character pointer variable named `FindGuess` that is initialized to `NULL`.

      a character array named `GuessALetterCopy` of size `MAX_INPUT` (`MAX_INPUT` is defined as 81)

      an `int` variable named `FoundALetter` that is initialize to 0

Copy `Phrase` into `GuessALetterCopy`.

Print `PhraseCopy`

Print `"Player 2 : Guess a letter : "`

Put their guess into the variable `Guess`.

Use `strchr()` to find `Guess` in `GuessALetterCopy` and store the pointer in `FindGuess`.

while `FindGuess` is not `NULL`

      Set `FoundALetter` to 1

      Use pointer arithmetic to find the difference between `FindGuess` and `GuessALetterCopy`. Use that distance to set the element in `PhraseCopy` to that same element from `Phrase`.

      `PhraseCopy[FindGuess - GuessALetterCopy] = Phrase[FindGuess - GuessALetterCopy];`

      Dereference `FindGuess` and set to a dash to prevent an infinite loop

      Use `strchr()` to find `Guess` in `GuessALetterCopy` and store the pointer in `FindGuess`.

      Return the value of `FoundALetter`

Step 3 – Create function `main()` in your `Code4.c` file

Create the following variables

      a character array named `Phrase` of size `MAX_INPUT`

a character array named `PhraseCopy` of size `MAX_INPUT`

a character array named `Alphabet` and initialize it to all upper case letters of the alphabet.

a character pointer named `ReplaceIt` and set it to `NULL`

an integer named `Strikes` and initialize to 0

an integer named `YourOut` initialized to 3

Print `"Welcome to "`

Call `ConvertDecimalToBinary()` with `YourOut`

Print `" STRIKES - YOU'RE OUT - the CSE version"`

Print `"Player 2 - Please look away"`

Print `"Player 1 - Please enter the phrase that Player 2 will be guessing. "`

Print `"          Enter a maximum of xxx characters. "` where xxx is `MAX_INPUT-1`

Print `"          Your phrase CANNOT contain a dash. "` where xxx is `MAX_INPUT-1`

Use `fgets()` to read the input into `Phrase`.

Use `strchr()` to determine if `Phrase` contains a dash. If it does, print `"You broke the rules.  We can't play. BYE!!"` and `exit()` the program.

Remove the `\n` from `Phrase`.

Use a `for` loop from 0 to the string length of `Phrase` and copy the uppercase version of each element of `Phrase` into `UpperPhrase`.

Copy `UpperPhrase` into `PhraseCopy`.

Use `strpbrk()` to find all characters from `Alphabet` in `PhraseCopy` and replace them with a dash (hint : `while` loop). Store the pointer returned by `strpbrk()` in `ReplaceIt`.

Print `"Player 2 - Here's the phrase you need to guess"`

Use a do-while `strchr()` can find a dash in `PhraseCopy`

      If calling `GuessALetter` with `Phrase` and `PhraseCopy` returns a 0, then a letter was not found

           Increment `Strikes`

           Print `"Strike "`

           Call `ConvertDecimalToBinary()` with `Strike` to print it in binary

           If `Strikes` is greater than or equal to `YourOut`, then

                 Call `ConvertDecimalToBinary()` to print `Strikes` in binary

                 Print `" STRIKES - YOU'RE OUT!!"`

                 Print `"Game over"`

                 Gracefully exit the program (DO NOT USE `exit()`)

Print "You figured it out!!"

Print "Player 1 entered the phrase"

Print Phrase

Print "Player 2 WINS!!!!"

Part 4 – makefile

Change your `makefile` to use your new `Code4.c` file. You are using the same name for the library so you should not need to change that in the `makefile`. Compile your program and run your program.

Part 5 - Testing

Run your `Code4.e` and confirm that your output matches the output in the assignment. Confirm that you have met all elements of the rubric.

Part 6 – Code Submission

Submit the following files

```
Code4_xxxxxxxxx.c

MyLib.c

MyLib.h

makefile
```

```
Welcome to 00000011 STRIKES - YOU'RE OUT - the CSE version

Player 2 - Please look away

Player 1 - Please enter the phrase that Player 2 will be guessing.
            Enter a maximum of 80 characters
            Your phrase CANNOT contain a dash

This is a TEST! 123

Player 2 - Here's the phrase you need to guess


---- -- - ----! 123

Player 2 : Guess a letter : T


T--- -- - T--T! 123

Player 2 : Guess a letter : I


T-i- i- - T--T! 123

Player 2 : Guess a letter : a


T-i- i- a T--T! 123

Player 2 : Guess a letter : h


Thi- i- a T--T! 123

Player 2 : Guess a letter : s


This is a T-ST! 123

Player 2 : Guess a letter : e


You figured it out!!

Player 1 entered the phrase

This is a TEST! 123

Player 2 WINS!!!!
```

```
Welcome to 00000011 STRIKES - YOU'RE OUT - the CSE version
```

```
Player 2 - Please look away

Player 1 - Please enter the phrase that Player 2 will be guessing.
            Enter a maximum of 80 characters
            Your phrase CANNOT contain a dash

This is a test - 123!

You broke the rules.  We can't play.  BYE!!
```

```
Welcome to 00000011 STRIKES - YOU'RE OUT - the CSE version

Player 2 - Please look away

Player 1 - Please enter the phrase that Player 2 will be guessing.
            Enter a maximum of 80 characters
            Your phrase CANNOT contain a dash

This is a TEST! 123

Player 2 - Here's the phrase you need to guess


---- -- - ----! 123

Player 2 : Guess a letter : T


T--- -- - T--T! 123

Player 2 : Guess a letter : i


T-i- i- - T--T! 123

Player 2 : Guess a letter : a


T-i- i- a T--T! 123

Player 2 : Guess a letter : e


T-i- i- a TE-T! 123

Player 2 : Guess a letter : i


T-i- i- a TE-T! 123

Player 2 : Guess a letter : o

Strike 00000001
```

```
T-i- i- a TE-T! 123

Player 2 : Guess a letter : u

Strike 00000010

T-i- i- a TE-T! 123

Player 2 : Guess a letter : l

Strike 00000011

00000011 STRIKES - YOU'RE OUT!!

Game over
```

```
Welcome to 00000011 STRIKES - YOU'RE OUT - the CSE version

Player 2 - Please look away

Player 1 - Please enter the phrase that Player 2 will be guessing.
          Enter a maximum of 80 characters
          Your phrase CANNOT contain a dash

Works for both UPPERCASE and lowercase.

Player 2 - Here's the phrase you need to guess


----- --- ---- --------- --- ---------.

Player 2 : Guess a letter : u


----- --- ---- U-------- --- ---------.

Player 2 : Guess a letter : p


----- --- ---- UPP------ --- ---------.

Player 2 : Guess a letter : c


----- --- ---- UPP--C--- --- -----c---.

Player 2 : Guess a letter : a


----- --- ---- UPP--CA-- a-- -----ca--.

Player 2 : Guess a letter : S
```

```
----s --- ---- UPP--CAS- a-- -----cas-.

Player 2 : Guess a letter : E


----s --- ---- UPPE-CASE a-- ---e-case.

Player 2 : Guess a letter : r


--r-s --r ---- UPPERCASE a-- ---ercase.

Player 2 : Guess a letter : n


--r-s --r ---- UPPERCASE an- ---ercase.

Player 2 : Guess a letter : d


--r-s --r ---- UPPERCASE and ---ercase.

Player 2 : Guess a letter : l


--r-s --r ---- UPPERCASE and l--ercase.

Player 2 : Guess a letter : o


-or-s -or -o-- UPPERCASE and lo-ercase.

Player 2 : Guess a letter : W


Wor-s -or -o-- UPPERCASE and lowercase.

Player 2 : Guess a letter : K


Works -or -o-- UPPERCASE and lowercase.

Player 2 : Guess a letter : f


Works for -o-- UPPERCASE and lowercase.

Player 2 : Guess a letter : B


Works for bo-- UPPERCASE and lowercase.

Player 2 : Guess a letter : T
```

```
Works for bot- UPPERCASE and lowercase.

Player 2 : Guess a letter : H


You figured it out!!

Player 1 entered the phrase

Works for both UPPERCASE and lowercase.

Player 2 WINS!!!!
```