Homework 6: Switch Statement Shenanigans

**Assigned**: March 28th, 2018 **Due**: April 4th, 2018

Make a subdirectory "hw6" of your cs240 folder for this assignment. Copy the starter files from /courses/cs240/s18/kamaral/GROUP/hw6 to your hw6 subdirectory.

**Part 1: Do While and Switch Case**

In the file onceplus.c, there is a function called process which takes one argument, a single integer, and returns an integer value.

There are also a number of symbolic constants defined at the beginning of the file.

The process function should include a switch case using the argument as it’s condition value.

For every symbolic constant at the beginning of the tile, there should be a separate case for that value. They should do the following:

* For ERROR\_404, the switch should print “Not Found.” on its own line.
* For ERROR\_403, the switch should print “Forbidden.” on its own line.
* For ERROR\_400, the switch should print “Bad Request.” on its own line.

Also, include a case for EOF.

* For EOF, the switch should print “End of Stream” and then return 0 or FALSE.

In the default case, print “Valid input found: ” and then the argument as an ASCII character followed by a newline character.

If the argument wasn’t EOF, then the process function should return 1 or TRUE.

In the main file, write a do while loop to read one character of input from standard input at a time and pass it to the process function as an integer. Since we’re letting the process function handle EOF, don’t cast the input down to an integer just yet.

Use the process function’s return value to decide if the loop should continue as the do while’s condition.

This program should only contain ONE (1) instance of the getchar function and NO OTHER INPUT FUNCTIONS!

In your typescript, test onceplus.c using the provided input file test.in. Your results should match test.out (do not overwrite this file).

**Part 2: Basic Calculator**

In the file calc.c, the main method is already written for you. The main method reads from standard input an expression of the following form:

<operand> <operator> <operand>

There is also a function called doMath which takes two integer arguments and a character argument and returns an integer value. The first two arguments are the operands collected from input in the same order as they appear. The last character argument is the ASCII representation of the math operator from input.

Your job is to finish the doMath function so that it supports the following operations:

* Addition (+)
* Subtraction (-)
* Multiplication (\*)
* Division (/)
* Remainder (%)
* XOR (^)
* Logical AND (&)
* Logical OR (|)

The function should return the mathematical result of the operation on the two operands.

Use a switch statement. This function can and should be done without the use of if statements. The default case is that the doMath function print an error message and then should call the function exit(EXIT\_FAILURE).

The exit function allows the program to terminate immediately from any point in the code and the argument to it is a status code. We’ll discuss this function further as we discuss larger projects. For now, just use it as is.

**Exercise: Practice makes perfect**

As an exercise, but not to be graded or anything, see if you can adapt the above code under the following restrictions:

* Do the whole program inside of the main, no extra functions.
* Read multiple lines of expressions and terminate on EOF or an error.
* Don’t use the exit(EXIT\_FAILURE) to close the program on unknown operator.
  + Hint: Consider the use of break or continue here!

This will not be graded, so I recommend you put this work in a new file like calc2.c or store it in your home directory.