4. PROGRAMMING ASSIGNMENT 4  UPDATED 10/21/17
Read: Chapter 7 (Skip Recursion: pages 158-160)
Programming: Name your program p4.c
DUE: Saturday, October 28, 2017 @ 6:00am

START EARLY!!!  This program should be modular, with logical tasks broken up into functions. In this assignment you will write a program using functions to simulate a simple integer calculator, draw 1 shape, and ascertain whether a prime number. Do NOT use global variables in CSE 5a. Example:

```
ENTER 2 whole numbers (3 5): 3 5
ENTER menu operation (+ - * / A B Q): +
3 + 5 is 8
ENTER 2 whole numbers (3 5): 4 7
ENTER menu operation (+ - * / A B Q): -
4 - 7 is -3
```

User first types in the two numbers: “3 5”

User next types in the expression: “+”
Three is added to Five and the total eight is displayed.

User first types in the two numbers: “4 7”

User next types in the expression: “-”
Four is subtracted from Seven and the total -3 is displayed.

a) main() allows the user to input an expression consisting of 2 operands and an operation.

```
printf("ENTER 2 whole numbers (3 5): ");
scanf("%d%d", &num1, &num2);                      // Input integers
getchar();                                         // Read <ENTER> key
printf("ENTER menu operation (+ - * / A B Q): ");
oper = getchar();                                  // Input operation
getchar();                                         // Read <ENTER> key
```

Valid character operators are: + - * / % A a B b Q q
Addition: +
Subtraction: -
Multiplication: *
Division: / (Error check for division by zero requires new input)
Modulus: % (Error check for division by zero requires new input)
Print box: A (a – Case insensitive)
Prime number: B (b – Case insensitive)
Exit program: Q (q – Terminate program with “exit(0);” #include <stdlib.h>)

Display an invalid operator error message if input character is other than these valid operators: + - * / % A a B b Q q

IF an invalid operator is entered, display error message, prompt to repeat valid (numeric and operator) input. A loop will repeat the prompt until a valid input is entered, thus menu() is called again.
Do not calculate with invalid operators. See sample output for example. Refer to “Decisions” handout.

Display an appropriate error message if the operator is '/' and the 2nd operand is zero for invalid division by zero. Prompt to repeat valid (numeric and operator) input. Do the same for modulus operator.
Grader will enter ONLY two integers. NO floating point is entered.

main() will call “ menu(); “ which displays the operations. main() reads and checks for valid input, displaying an error message if menu choice is invalid. If menu choice is valid, then call the appropriate function, add(num1, num2);
main() allows input of operator "Q" to quit the program.
```
//e.g. printf("QUIT:\n\"Q\" OPERATORS::\n" + - * / %\% A B\n\"n\");  
```

Note: \" is used in printf to display the actual double quotation mark
\% is used in printf to display the actual percent symbol
Display a friendly exit message. Use “exit(0);” for program termination and “ #include <stdlib.h>”. Refer to “Decisions” handout.
You will write 8 functions besides `main()`. `main()` will call (invoke) all 8 functions. The function declarations are:

```c
void menu(void);
int add (int n1, int n2);
int sub (int n1, int n2);
int mult (int n1, int n2);
int divide(int n1, int n2);
int mod   (int n1, int n2);
void prtBox (int n1, int n2);
int isPrime(int n);
```

```c
int main( void )
{
    char oper;  // Arithmetic operator ...
    do
    {
        menu();
        printf("ENTER 2 whole numbers (3 5): ");
        scanf("%d%d", &num1, &num2);  // Input integers
        getchar();                    // Read <ENTER> key
        printf("ENTER menu operation (+ - * / % A B Q): ");
        oper = getchar();             // Input operator
        getchar();                    // Read <ENTER> key
        switch( oper )
        {
            case '+': total = add(num1,num2);
                        printf("%d + %d is %d
", num1, num2, total);
                        break;
            case 'B': if( isPrime(num1) )  // Is a prime number?
                        printf("%3d is a prime number
", num1);
                        else
                        printf("%3d is not a prime number
", num1);
                        // Repeat for num2 ...

        }
    }while( /* Repeat while not a Q nor q */ );
}
```

b) `menu()` is called from `main()` and uses a series of printfs to display a menu of choices.

e.g. `printf("   +   ADD \
"
        "   %   MODULUS \
"`;

c) In `main()`, determine the menu choice to call the appropriate function

e.g. "total = add(num1, num2);
        printf("%d + %d is %d\n", num1, num2, total)";

d) `add()` is passed the two operands, num1 and num2. `add()` is called from `main()`. Return the sum and print the total in `main()`.

e) Repeat step d) with operator '-' to call "sub()" function for subtraction of num2 from num1. Return the difference and print the total in `main()`.

f) Repeat step d) with operators '*' to call "mult()" function for multiplication of num1 with num2. Return product and print the total in `main()`.

g) Repeat step d) with operator '/' to call "divide()" function for division of n1 by n2. Return total and print the total in `main()`. Note: In step a), `main()` checks for invalid zero operand values on division operator.

h) Repeat step d) with operator '%' to call "mod()" function for modulus of n1 by n2. Return new total and print the total in `main()`. Note: In step a), `main()` checks for invalid zero operand values on modulus operator. Since the modulus operator is defined only to work with integers in C, then type integer was chosen for operand and total.
i) Repeat step d) with choice 'A' (or 'a') to call “prtBox()” function for printing n1 rows by n2 columns.

j) Choice 'B' (or 'b') will first call “isPrime()” function passing first integer. Test if a prime number returning zero if false and one if true. Print result. Repeat with a second call and the second integer. Refer to Decisions handout for “foundPrime” flag variable.

Note: A Prime Number can be divided evenly only by 1, or itself. And it must be a whole number greater than 1. Example: 5 can only be divided evenly by 1 or 5, so it is a prime number. But 6 can be divided evenly by 1, 2, 3 and 6 so it is NOT a prime number (it is a composite number).

k) Your program must loop so that the user can continue to generate and view the output until the user responds with “Q” (or q – case insensitive)

l) Every function (including main() ) REQUIRES a header comment. See page 5 of Lab Guide.

e.g. /*
   * Division of n1 by n2 parameter. Return total.
   * Not called if n2 is zero input
   */

HINT: Solve this problem in small steps. Here's a suggestion.
2) Write the code for step b).
3) Write the code for step c). Test.
4) Write the code for steps d) and e) and f). Test.
5) Write the code for step g). Test. Do step h).
7) Write the code for step k) so your entire program will loop again. Test.
8) Complete code for step l) and a) to check your input. Test. Test. Test. DONE!

PA#4 SAMPLE OUTPUT (SAMPLE INPUT is typed in bold below)

<table>
<thead>
<tr>
<th>CALCULATOR OPERATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>+    ADD</td>
</tr>
<tr>
<td>-    SUBTRACT</td>
</tr>
<tr>
<td>*    MULTIPLY</td>
</tr>
<tr>
<td>/    DIVIDE</td>
</tr>
<tr>
<td>%    MODULUS</td>
</tr>
<tr>
<td>A    PRINT BOX</td>
</tr>
<tr>
<td>B    PRIME NUMBER?</td>
</tr>
<tr>
<td>Q    QUIT!</td>
</tr>
</tbody>
</table>

=======================================
ENTER 2 whole numbers (3 5):
3 5
ENTER menu operation (+ - * / % A B Q):
3 + 5 is 8

<table>
<thead>
<tr>
<th>CALCULATOR OPERATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>+    ADD</td>
</tr>
<tr>
<td>-    SUBTRACT</td>
</tr>
<tr>
<td>*    MULTIPLY</td>
</tr>
<tr>
<td>/    DIVIDE</td>
</tr>
<tr>
<td>%    MODULUS</td>
</tr>
<tr>
<td>A    PRINT BOX</td>
</tr>
<tr>
<td>B    PRIME NUMBER?</td>
</tr>
<tr>
<td>Q    QUIT!</td>
</tr>
</tbody>
</table>

=======================================
ENTER 2 whole numbers (3 5):
3 5
ENTER menu operation (+ - * / % A B Q):
3 - 5 is -2
NOTE: Omitted entire menu output

CALCULATOR OPERATIONS
=======================
ENTER 2 whole numbers (3 5): 3 5
ENTER menu operation (+ - * / % A B Q): *
3 * 5 is 15

CALCULATOR OPERATIONS
=======================
ENTER 2 whole numbers (3 5): 3 5
ENTER menu operation (+ - * / % A B Q): /
3 / 5 is 0

CALCULATOR OPERATIONS
=======================
ENTER 2 whole numbers (3 5): 3 5
ENTER menu operation (+ - * / % A B Q): %
3 % 5 is 3

CALCULATOR OPERATIONS
=======================
ENTER 2 whole numbers (3 5): 3 5
ENTER menu operation (+ - * / % A B Q): A
*****
*****
*****

CALCULATOR OPERATIONS
=======================
ENTER 2 whole numbers (3 5): 5 20
ENTER menu operation (+ - * / % A B Q): b
5 is a prime number
20 is not a prime number

CALCULATOR OPERATIONS
=======================
ENTER 2 whole numbers (3 5): 3 0
ENTER menu operation (+ - * / % A B Q): %
ERROR - division by zero!

CALCULATOR OPERATIONS
=======================
ENTER 2 whole numbers (3 5): 3 8
ENTER menu operation (+ - * / % A B Q): Z
ERROR! Invalid Operator.

CALCULATOR OPERATIONS
=======================
ENTER 2 whole numbers (3 5): 1 2
ENTER menu operation (+ - * / % A B Q): q
BYE!!

Submit the final version of your program as p4.c

Verify you SAVED your work in the Documents - cs5f HOME directory.