Introduction to C - Programming Assignment #9

Objective
1. To give students practice in writing and calling their own functions.

Problem: Educational Software
Your little brother is having trouble with arithmetic. Your parents realize that after taking a few weeks of your C programming course, that you could potentially write a computer program that will allow him to practice his arithmetic skills.

In particular, your program will allow your brother to play two separate games:

1) A game where he has to complete several additions or multiplications.

2) A game where he has to determine a secret number after being told if his guesses are too high or too low.

Your program should prompt your brother with the following menu:

1) Play Arithmetic Game
2) Play Guessing Game
3) Print Score
4) Quit

If he chooses option 1, then you should prompt him with the following menu choices:

1) Addition
2) Multiplication

Your program should then prompt him for the maximum value of the numbers to be used in the problems. You will ask 10 questions where each operand is in between 1 and this maximum value.

If he chooses option 2, your program should generate a random integer in between 1 and 100, inclusive. After that, it should prompt your brother for his first guess. After each guess, your program should tell him whether to guess higher or lower. This continues until he gets the number exactly. To calculate his score, compute 17 minus the number of guesses he made. If this number is in between 0 and 10, inclusive, this is his score. If the number is less than 0, give him a score of 0. If this number is greater than 10, give him a score of 10.

For option 3, simply report your brother's total score, which is the sum of his scores from each round he plays.

Implementation Details

You will be required to write four functions with the prototypes given below. (Note: you may write other functions as well, but these four are required.) Your functions should do what the comments for them below specify:

```
/*Pre-Conditions: This function takes in two integers, max
and op. This function gives the user 10 arithmetic
questions, where each operand ranges from 1 to max,
inclusive. The value of operator dictates whether the
problems are addition or multiplication problems. Namely,
if op is 1, they are addition problems, otherwise, they are
multiplication problems.

*Post-Conditions:  The function returns the number problems
solved correctly, and prints out for the user after their
answer whether or not they got the question correct.
*/
int arithGame(int max, int op);

/*Pre-Conditions: This function does not take in any
parameters. This function allows the user to play the
guessing game where the randomly generated number lies in
between 1 and 100, inclusive.

*Post-Conditions:  The value returned is the score of the
user in the game. This score is 17 minus the number of
guesses unless this value is lower than 0 or greater than
10. In these cases, 0 and 10 are returned, respectively.
*/
int guessGame();

/*Pre-conditions: Both parameters, a and b are integers.

Post-Conditions:  The  larger  of  the  two  parameters  is
returned.
*/
int max(int a, int b);

/*Pre-conditions: Both parameters, a and b are integers.

Post-Conditions:  The  smaller  of  the  two  parameters  is
returned.
*/
int min(int a, int b);
```

Seed the random number generator at the beginning of your program. Do this exactly once. Here is the line of code:

```
srand(time(0));
```

In order to use this you need to include stdlib.h and time.h at the top of the program.

Please use the following constants for ADD and MULT

```
#define ADD 1
#define MULT 2
```

Deliverables
A single source file named *game.c* turned in through WebCourses.

Restrictions
Although you may use other compilers, your program must compile in gcc and run in Code::Blocks. Your program should include a header comment with the following information: your name, course number, section number, assignment title, and date. Make sure you include comments throughout your code describing the major steps in solving the problem. **Make sure to use good programming style, including use of appropriate constants, good variable names and good use of white space. A significant portion of your grade will be based upon programming style and not correctness. Of course, a significant portion of your grade will also be based upon correctness.**

Sample Output
```
Please make a selection from the following:
1. Play Arithmetic Game.
2. Play Guessing Game.
3. Print Score.
4. Quit.
1
Would you like, 1)Addition or 2)Multiplication?
1
What is the maximum number you would like?
100
What is 21+86?
107
Correct, great job!
What is 87+96?
173
Sorry, that's incorrect, the answer is 183.
What is 86+70?
156
Correct, great job!
What is 55+4?
```

59
Correct, great job!
What is 13+17?
30
Correct, great job!
What is 89+73?
162
Correct, great job!
What is 22+11?
33
Correct, great job!
What is 67+9?
76
Correct, great job!
What is 35+94?
129
Correct, great job!
What is 16+85?
101
Correct, great job!
Your score for the round is 9.

Please make a selection from the following:
1. Play Arithmetic Game.
2. Play Guessing Game.
3. Print Score.
4. Quit.
2
Enter the guess!
50
Your guess is too high, try again.
Enter your guess!
30
Your guess is too high, try again.
Enter your guess!
10
Your guess is too high, try again.
Enter your guess!
5
Your guess is too low, try again.
Enter your guess!
7
Your guess is too high, try again.
Enter your guess!
6
Great, you guessed the correct number 6 in 6 guesses.
Your score for the round is 10.

```
Please make a selection from the following:
1. Play Arithmetic Game.
2. Play Guessing Game.
3. Print Score.
4. Quit.
3
Your score is 19.

Please make a selection from the following:
1. Play Arithmetic Game.
2. Play Guessing Game.
3. Print Score.
4. Quit.
4
Thank you for playing!
```