C Programming Final Exam

This is a take-home final. Click the Final Exam Link to **submit**, just like an assignment.

Answer all questions and follow the rules below.

- Please send one single text file (not Word, a Zip file, or other formats) with all your code.

- Name your single submitted file using the following format: [Your Last Name]_final.txt

... so if your last name was Jones, your submitted file would be called: Jones_final.txt

- Do not send multiple files!

- Do not include hyperlinks, paste in the code

... and again, all in one single text file. Do not email it to me, rather, attach a single text file to the Final Exam submission link just like you normally do with a homework assignment.

While it is important for you to test out the answers with your favorite compiler, just put the actual answers in your submission. For questions 1-7, that means just the macro or function (or set of functions) that can be called to answer the question (I don't need the main function or sample output). Most questions that request a function can be easily answered within a single function.

For Question 8, just need a set of functions that you would call ... no need for a main program that calls them. It is best to create a function for each baseball statistic (one for Totals Bases, Batting Average, etc.) as that is what I am looking for with this question.

For Question 9, just provide the list of structures and their members/types as needed.

Answer questions just as you do on your homework assignments (this means comments, headers, coding standards ...).

This is a chance for you to show me what you learned. Make your answer the BEST it can be. I am not going to tell you how to solve these problems. Again, just getting it to work does not guarantee the highest mark.

Grading **will be based** on readability, maintainability, efficiency, and re-usability in addition to correctness.

Good Luck!

Tim

The Questions:

1) Write a **macro** for each of the following. DO NOT provide a function ... I am looking for a macro (see last set of lecture notes).

   a) Area of a Circle
   b) Area of a Square
   c) Area of a Rectangle
   d) Area of a Triangle

You can find formulas and detailed explanations at: http://www.mathisfun.com

2) Write a **function** which will determine how many words are in a given string. You can assume that one or more consecutive white spaces is a delimiter between words, and that the string you pass to your function is null terminated.

3) Write a **function** that is passed a *month, day, and year* and will determine if that date is valid. You can assume each parameter is passed in as an integer. Remember to check for leap year!

```
validDate (5, 31, 1961)    .... would be valid

validDate (13, 4, 1967)    ... would be invalid, the month is invalid
```

4) Write a function that takes the values of a two-card blackjack HAND as input, and returns the point total of the hand. The value of the cards '2' through '9' is equal to their face value, the cards 'T', 'K', 'Q', 'J' are worth 10 points and the ace ('A') is worth 1 unless it comes with another ace, then that second ace is worth 1 point. The program should be able to catch incorrect input.

```
Enter cards: A Q
    The score is 21

Enter cards: A A
    The score is 12

Enter cards: T 7
    The score is 17

Enter cards: A 5
    The score is 16

Enter cards: 7 #
     *** Would be invalid, # is not a valid card

Enter cards: Z 4
     *** Would be invalid, Z is not a valid card
```

Hint: I've used a value of 'T' for the 10 card so you can simply pass in two characters, instead of strings, as parameters to this function.

5) Write a **function** to determine is a given word is legal. A word is illegal if it contains no vowels. For this problem, the letter Y is considered to be a legal vowel. Pass in a word to this function and it will determine if the word is legal or not. You can make the following assumptions about the word you a passing to this function.

1) The string being passed is a combination of letters only (no non-letter check needed)
2) The string being passed is null terminated
3) Letters may be capital or lower case and it has no effect on whether its a word

Examples:

sch - is illegal, no vowels
apple - legal, contains a vowel
APPle - legal, uppercase letter combinations do not matter
try - legal, no vowel, but contains the letter 'y'

6) Write a **function** that will determine if a given string is a palindrome. DO NOT use the C library function: strrev

A *palindrome* is a word or sentence that reads the same forward as it does backward.

Examples of word palindromes would be *civic* or *rotor* ... a word or phase would be:

Never odd or even

A good web site of examples is: *http://www.rinkworks.com/words/palindromes.shtml*

7) Write a **function** that will return in a structure the following characteristics of a given string:

1) string length (use strlen),
2) number of upper case characters
3) number of lower case characters,
4) number of digits
5) number of blank spaces
6) number of non-alphanumeric characters.

8) Develop a **set of function(s)** to compute various offensive statistics on baseball. The following information will be available on each player:

Number of *Walks (BB), Strike Outs (SO), Hit by Pitch (HP), Sac Flys (SF), Singles, Doubles (2B), Triples (3B),* and *Home Runs (HR)* as well as *Number of At Bats (AB).*

Based on this information, develop **a set of functions** that will compute the following:

Total Bases, Batting Average, Home Run Ratio, Strike Out Ratio, On Base Percentage, and *Slugging Average.*

You do not need to be a baseball fan to do this ... All the information you need in terms of the formulas and explanations can be found at:

http://www.baseball-almanac.com/stats.shtml

Note: ***Number of hits*** is: singles + doubles + triples + home runs

If you want to test if your functions are working, see compiled stats of the 2014 Boston Red Sox at: http://www.baseball-reference.com/teams/BOS/2014.shtml

Note: All I am looking for here is a *set of functions* ... you do not need to put them into a program and show me how you call them, i.e., no main function needed.

9) Most people enjoy watching movies these days, whether its the classics or modern ones. **Develop a set of structures** that could be used to model the information about a movie collection. What type of information would you want to collect and store about a movie? What would be the right types in C for that information? Define supporting structures as needed and have one final structure type that is made up of various members (some members may be on some structure type, others may be simple integers, floats, chars, arrays, etc).

No program is needed although you might want to create a simple main function and include your structure types just to test that everything compiles.

This question is similar to the final question on the midterm, but you have learned about many different types since then. Here is a template to use to get started and indicates what I am looking for in your answer. Use everything you learned this semester, especially types from the last set of lectures notes.

```
/* add supporting structures - expect many structure types here ... date is good example */


struct date
{
      int month;
      int day;
      int year;
};
```

```
/* add other supporting structures */


/* Final structure, such as struct movie */
     struct movie
     {

/* some members may be a structure type themselves, here is an example */
            struct date releaseDate;   /* the date the movie was released */


/* other members may be ints, floats, doubles, chars, enum, ... */
            char title [100];   /* the title of the move */


/* add others */

     };


     int main  (  )
     {
            struct movie myMovie [1000];

/* nothing else needs to be added to main */


            return (0);

      };
```