

C Programming Midterm

Directions:

This is the midterm for the C Programming Class.

Answer electronically the questions and submit them into BlackBoard Learn by the Due Date.

Answer all questions and follow the rules below.

Please attach one single text file (not Word or other formats) with all your answers.

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

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

Do not send multiple files!

I just need the answer ... do not send program output, only the C source code ... and again, all in one single text file.

Understand that if I ask for a ***function***, all I am looking for is that function in your answer ... not a complete program that includes a main function ... I just need to see your function as the answer.

If I ask for a ***program***, then I am looking for a main function that calls one or more functions (include the functions).

If I ask for a ***code segment***, then all I am looking for is just the code needed to answer the question.

Rules:

1. You may use your book and notes
2. Quality, Re-usability, Maintainability, Readability, and Efficient code will be scored in addition to correctness.
3. Same coding rules and standards apply as your homework assignments (comments, headers, style, ...)
4. Use your computer to compile your code and test it out
5. Don't converse with other students or friends
6. Contact me if you have questions or state your assumptions

Questions:

- 1) Use the link below for the background needed to create **two functions** that work with a Triangle

<https://www.mathsisfun.com/triangle.html>

- a) Write a function that will return the area of a triangle
- b) Write a function that will return the perimeter of a triangle

2) Write a **function** `int frequency (int theArray [], int n, int x)`

that counts the number of times the item x appears among the first n elements of theArray and returns that count as the frequency of x in theArray. Use integers for everything!

For example, if the array being passed contained the values 5, 7, 23, 8, 23, 67, 23

... and n was 7 and x was 23, then it would return a value of 3 since 23 occurs 3 times within the first 7 elements of the array.

3a) (**Function**) Represent the mathematical operation: $\max(v, w, x, y, z)$ as a C function. This is, write a **function** that is passed 5 integer values that returns the largest value of those four parameters. For example, $\max(5, 2, 7, 4, 1)$ would return 7 since it is the largest value of the five parameters.

3b) (**Code Segment**) Rewrite the mathematical operation: $\max(v, w, x, y, z)$ as

a single statement using the **conditional expression operator**.

For example, $\max(x, y)$ would be:

```
max = (x > y) ? x : y;
```

Hint: You will have more than one conditional expression operator (?) in your answer, and it would help to use boolean logic operations that were covered in Week 3.

4) Given the following test scores and grade equivalents, write a **function** which is passed a score, and returns a letter grade based on the score entered. It should also check for invalid values (a number less than 0 or greater than 100) and return an 'I' in that case.

Score	Grade
-----	-----
90-100	A
80-89	B
70-79	C
60-69	D
0 -59	F

5) Write a **function** that is passed an array of characters containing letter grades of A, B, C, D, and F, and prints a report that shows the total number of occurrences of each letter grade. Your function should accept both lower and upper case grades, for example, both 'b' and 'B' should be bucketed into your running total for B grades. Any grade that is invalid should be bucketed

as a grade of 'I' for Incomplete.

You must use a switch statement, and your function should accept an array of any size. Feel free to pass in the array size as parameter so you know how many grades you'll need to check in your loop.

For example, if you passed a function the following array:

```
char grades [ ] = {'A', ' ', 'b', 'C', 'x', 'D', 'c', 'F', 'B', 'Y', 'B', 'B', 'A'};
```

It would print:

Grade	Total
-----	-----
A	2
B	4
C	2
D	1
F	1
I	2

6) It's a World Cup year! To celebrate, FIFA (Fédération Internationale de Football Association) has asked you to write a **program** that will calculate various offensive and defensive statistics.

To get started, go to the following URL to familiarize yourself with various common soccer statistics.

<https://www.meracalculator.com/sports/football-statics.php>

Below are a combination of formulas to determine four common Defense and three Offense statistics. Your job here is to **develop and call functions** that when passed specific information, will **return a information within a structure**. FIFA would like at least two functions, one that will calculate and return defensive statistics and another similar one of offensive statistics.

Defense

Goalie Games Won Percentage = Games Won / (Games Won + Games Lost)

Goalie Saves = Shots on Goal - Goal Scored

Saves Percentage = Goalie Saves / Shots on Goal

Saves Ratio = Shots On Goal / Goalie Saves

Offense

Scoring Percentage = (Scoring Attempts - Goals Scored) / Scoring Attempts

Scoring Ratio = Shots On Goal / Goals Scored

Games Won Percentage = Team Games Won / (Team Games Won + Team Games Lost)

Review the example in it that provides the following inputs which I pasted below. This will provide test data and an idea of what to pass to your

functions.

Defense team:

Shots on Goal = 5

Goals Scored = 2

Games won in Goal = 1

Games Lost in Goal = 1

Offense Team:

Scoring Attempts (shots on goal) = 7

Goals Scored = 3

Team Games Won = 3

Team Games Lost = 4

In summary, write a **program** that has **at least two functions** that can be called. Use the information above to determine the parameter types and information returned in a structure for each function. Call these functions from a main function using the example information above. No need to prompt for any values, for this exam, just pass in the values. You do not have to worry about printing the information out, you are welcome to do that on your own to verify your work, but it is not required and will not be factored into the scoring of this question.

7) Write a **function** that raises an integer to a positive integer power. Call the function `x_to_the_n`, taking two integer arguments `x` and `n`.

Have the function return a long int, which represents the results of calculating `x` to the `nth` power. Do not use the C `pow` library function.

Example: if `x = 3` and `n = 2`, the function would return 9.

NOTE: Do not use recursion (since that version is in your lecture notes).

8) (Code Segment)

Provide the code need to DECLARE an array of structures that could store information about a group of passports. Do not write a program or worry about initializing the structure, all I am looking for is a set of structure types whose combined members would hold all the information found in a Passport, AND any supporting structures. Be careful on how you declare the members of a structure. Grading will be based on the flexibility of your design (i.e., define many structures and have structures within structures if necessary).

To figure out what information you need to collect, search on line for sample images of US Passports. NOTE: Don't go overboard such as having a structure with only one member just to create many structures.