DEPARTMENT OF
SOFTWARE TECHNOLOGY

Hands-On Exam 1: The Strange Grade

A student transferred sections for a CCPROG1 class. He only has 3 quizzes left and 2 hands-on activities that has not been graded. This student also has an exemption for the third quiz which will receive the average of the first 2 missing quizzes. To help him compute his grades properly, he wrote some software to help him compute his grades.

For this problem, complete the requirements in the template file provided. The problem is worth 50 points in total.

encodeQuizzes (10pts)
This function takes in three pointers (pQuiz1, pQuiz2 and pQuiz3) as parameters. It first asks the user to input 2 floating point numbers and assign the values to the first two parameters, pQuiz1 and pQuiz2. It then takes the average of the user inputs and saves it via the third parameter pQuiz3. There should be no user prompts or display (print) statements in this function. Assume the user will always input 2 valid floating point numbers.

For the scanning of inputs, the following are samples of valid and invalid prompting.

Incorrect prompting			Correct prompting

`Input grade 1: 91.5`
`Input grade 2: 88.0`
	`Grade 1: 91.5`
	`88.0`
		`Grade inputs`
		`91.5`
		`88.0`
			`91.5`
			`88.0`

encodeHandsOn (10pts)
This function takes in two pointers (pHO1 and pHO2) and extracts the values based on one integer input from the user. Assume the user will always input a 4 digit number.The function will ask for a 4 digit integer number to the user. It will assign the first two leftmost digits as the value for fHO1 and the two rightmost digits as the value for fHO2.

For the scanning of inputs, the following are samples of valid and invalid prompting.

Incorrect prompting			Correct prompting
`Input HO1: 92` `Input HO2: 87`	`Input HO: 9287`	`Input HO` `9287`	`9287`

This is an example of how the input number is split into the 2 hands on values.

Input Number	*pHO1	*pHO2
9287	92	87
8377	83	77
6789	67	89

computeDiminishingTotal (10pts)
This function takes in the grades for the activities, midterms, hands-on exams, final exam and final project and will compute the total grade. The total grade is the product of the percent equivalent of the grades. The final total grade should be converted back from the percent grade.

Formula:
Total grade = (Activities in %) * (Midterms in %) * (hands-on exams in %) * (final exam in %) * (final project in %) * 100

Example of a percentage equivalent:

Grade Fed	Grade as a Percentage
89.75	0.8975
96.0	0.960
86.0	0.860
88.5	0.885
92.5	0.925

Example computation:

Total grade = 0.8975 * 0.960 * 0.860 * 0.885 * 0.925 * 100 = 60.66

isFail (10pts)
This function checks if the grade fed is a failing grade or not. Should the grade be below 60, the function returns the value 1 and will return the value 0 otherwise. This function is to be written without the use of conditional constructs. The use of conditional constructs is not permitted and a grade of zero will be given for the problem should conditional constructs be used.

Hint: It is probably easier to check if the grade passes first then modify that condition to check for a failing mark.

isMPLowest (10pts)
This function takes in the grades for the activities, midterms, hands-on exams, final exam and final project and will determine if the project grade is the lowest grade among the given grades. The function should return 1 if the project variable is the lowest value and 0 otherwise. This function is to be written without the use of conditional constructs. The use of conditional constructs is not permitted and a grade of zero will be given for the problem should conditional constructs be used.

Hint: Use the and (&&) operation to simplify the checking.

Sample Runs

The code shown are sample runs when running the HO1_P2.c (Text in red are the user inputs.)

Sample Run 1	Sample Run 2
91.5 88.9 8884 Results Quiz : 91.50 88.90 90.20 HO1&2 : 88 84 Total : 49.43 Fail : 1 MP low: 1	75.5 68.5 8092 Results Quiz : 75.50 68.50 72.00 HO1&2 : 80 92 Total : 39.46 Fail : 1 MP low: 0
Sample Run 3	**Sample Run 4**
99.9 99.9 9999 Results Quiz : 99.90 99.90 99.90 HO1&2 : 99 99 Total : 63.02 Fail : 0 MP low: 1	80.0 69.0 8987 Results Quiz : 80.00 69.00 74.50 HO1&2 : 89 87 Total : 41.77 Fail : 1 MP low: 0

Additional Instructions

- Only modify and submit the file **LASTNAME-Grade.c** for this problem in the AnimoSpace page.
- Please encode the student's name in the Author and Section field at the top of the file.
- Be sure to not modify the function headers.
    - Do not add additional parameters or change the return type.
- If the code does not compile, the problem will receive a grade of 0.0.

You are given two files for this problem:

1. LASTNAME-Grade.c which contains some initial code that you'll need to complete. Comments indicate the requirement and restrictions imposed for the solution. For this file, the student's implementation should not include any printf(). This file should not be modified to contain the main().
2. HO1_P2.c which contains the main() that can be used to test the requirement.

DELIVERABLE: Your C program source file **LASTNAME-Grade.c** with your own last name as filename. For example, if your lastname is SANTOS, then the source file should be named as SANTOS-Grade.c. Upload your source file in AnimoSpace before the indicated submission time.

TESTING & SCORING:

- Your program will be compiled via gcc -Wall. Thus, for each function that does not compile successfully, the score for that function is 0.
- Your program will be tested by your instructor with other main() (which may contain different values from the ones given to you) and with function calls of different parameter values.
- Full credit will be given for the function only if the student's implementation are all correct for all the test values used by the instructor during checking AND only if the student's implementation complied with the requirement and did not violate restrictions. Deductions will be given if not all test cases produce correct results. No credit is given if restrictions were not followed.