Task 1 The VOL(x, y, z) function used in puzzle1() does not compute the expected answer. The macro is given the arguements (1+1, 2+2, 3+3) which produces 12. We expect the product of the arguments given to be 48. However VOL does not produce this result because macros only do textual replacement with their arguments. Therefore when we give the compiler VOL(1+1, 2+2, 3+3) it interprets the result of that macro function as 1+1*2+2*3+3 = 12. If we want the result to be 48 we have to give VOL the arguments as ((1+1), (2+2), (3+3)).

Task 2 puzzle2() fails to swap the given values because int swap() changes the values of the pointers it is passed in the scope of it's own defintion, and does not swap the values stored at those pointers.

Task 3 There are not any explict errors in puzzle3(). However the function is unable to correctly add one to an int the size of INT-MAX. Therefore I added an error message to stop that from being a valid case.

Task 4 In test-safe-add(), the compiler raises warnings about a lack of parentheses around the operands of the bitwise operators. However, the presence or lack of the compiler reccomended paretheses does not affect output. The bigger issue is that in the definition of test-safe-add(), in order to return, the function has to add the two parameters it is testing, and use overflow to determine if that was a valid addition. However, overflow is an undefined behavior, meaning we can't rely on it to tell us anything. Therefore the basis by which we are deciding two integers can be added together is flawed. Therfore I changed the method of testing whether two integers can be added togther to not rely on test addition and overflow but rather on comparison of the parameters to known safe values and additions.

Task 5 In test-safe-div(), the compiler raises a warning because the function only uses one of the parameters it is passed. Therefore there is an unused parameter. The warning is raised by the compiler to warn us that there may be something we left out of our code. In other words, the compiler thinks we made a mistake, so it tells us we might want to double check whether we left something out or not. However, if we did not leave anything out that otherwise would be instrumental to the outcome of our code, the unused parameter should not be an issue. This is the case in test-safe-div where the test of whether a case of integer division is safe or not only relies on whether the divisor (second parameter) is 0 or not. Therefore the other argument should not be used, is not used, and is only there for code clarity. Nothing is wrong with puzzle5().

Task 6 The function set-vector() prevents the code from compiling because it uses an undeclared variable. In addition, the compiler also tells us that the fucntion contains an unused variable in its definition. These two things are related. The unused varibale in the definition of the function is supposed to be used in the place of the unitialized variable that prevents the code from compiling. Once that is fixed the code compiles. However there is one more error in the code. The function does fill the given array with a given value, but it also assigns the given value to the next spot in memory immeditatley following the location of the given array, as the loop used to fill the array runs from index 0 to index n, not index n-1. This bug can corupt memory in other parts of the program so it must also be fixed to ensure everything works as expected, not just the function in question. I also altered the puzzle6() function to print the entire array.

Policies

1 If you submit an assigment more than two days late (as in this hypothetical), you will recieve a 0 regardless of how many late days you have left.

2 This ones a guess because I couldn't find the answer: regrade requests should be submitted to your TA. They defintiely should be submitted within five days of recieving a grade/feedback.

3 Jon and Daenerys have broken the collaboration policy. Collaboration is strictly not allowed (without permission). Jon should have gone to office hours himself. Jon should also not have sex with his half sister but shit happens bruh.

4 When the floss just too good.