|  |
| --- |
| 1. Show the output produced by each of the following program fragments. Assume that i, j, and k are int variables. 2. 1 2 3. 0 4. 1 5. 0 |
| 1. If i and j are positive integers, does (-i) / j always have the same value as –(i/j)? Justify your answer.  In C89, the result of expression (-i) / j is implementation-defined behavior. so, we don’t know exactly what will happen. the result of division will be rounded up or down. In C99, however, the result of a division is always truncated toward zero. |
| 1. What is the value of each of the following expressions in C89? (Give all possible values if an expression may have more than one value.) 2. 1 3. -1 or -2 4. -1 or -2 5. 1 |
| 1. Repeat Exercise 3 for C99 2. 1 3. -1 4. -1 5. 1 |
| 1. What is the value of each of the following expression in C89? (Give all possible values if an expression may have more than one value.) 2. 3 3. -3 or 2 4. 3 or -2 5. -3 |
| 1. Repeat Exercise 5 for C99 2. 3 3. -3 4. 3 5. -3 |
| 1. The algorithm for computing the UPC check digit ends with the following steps: ... Why doesn’t this technique work? |
| 1. Would the upc.c program still work if the expression 9 – ((total – 1) % 10) were replaced by (10 – (total % 10)) % 10? Answer: If total is a multiple of 10, program will not work. |
| 1. Show the output produced by each of the following program fragments. Assume that i, j, and k are int variables. 2. 63 8 3. 3 2 1 4. 2 -1 3 5. 0 0 0 |
| 1. Show the output produced by each of the following program fragments. Assume that i and j are int variable: 2. 12 12 3. 3 4 4. 2 8 5. 6 9` |
| 1. Show the output produced by each of the following program fragments. Assume that i, j, and k are int variables: 2. 0 1 3. 4 10 6 4. 0 8 7 5. 3 4 5 4 |
| 1. Show the output produced by each of the following program fragments. Assume that i, j, and k are int variables: 2. 6 16 3. 6 -7 4. 6 23 5. 6 15 |
| 1. Only one of the expressions ++i and i++ is exactly the same as (i+=1); which is it? Justify your answer. ++i is the same as (i+=1), because ++i increase 1 immediately like an (i+=1), and also use to i + 1 in the expression. |
| 1. Supply parentheses to show how a C compiler would interpret each of the following expressions. 2. (((a \* b) – (c \* d)) + e) 3. (((a / b) % c) / d) 4. ((((-a) – b) + c) – (+d)) 5. ((a \* (- b)) / c) – d |
| 1. Give the values of i and j after each of the following expression statements has been executed. (Assume that i has the value 1 initially and j has the value 2.) 2. i = 3, j = 2 3. i = 0, j = 2 4. i = 1, j = 2 5. i = 1, j = 3 |