

```
(d) i = 1; j = 2; k = 3;
    printf("%d", (i + 5) % (j + 2) / k);
```

- W *2. If i and j are positive integers, does $(-i)/j$ always have the same value as $-(i/j)$? Justify your answer.
3. 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.)
- (a) $8 / 5$
 - (b) $-8 / 5$
 - (c) $8 / -5$
 - (d) $-8 / -5$
4. Repeat Exercise 3 for C99.
5. 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.)
- (a) $8 \% 5$
 - (b) $-8 \% 5$
 - (c) $8 \% -5$
 - (d) $-8 \% -5$
6. Repeat Exercise 5 for C99.
7. The algorithm for computing the UPC check digit ends with the following steps:
 Subtract 1 from the total.
 Compute the remainder when the adjusted total is divided by 10.
 Subtract the remainder from 9.
 It's tempting to try to simplify the algorithm by using these steps instead:
 Compute the remainder when the total is divided by 10.
 Subtract the remainder from 10.
 Why doesn't this technique work?
8. Would the `upc.c` program still work if the expression `9 - ((total - 1) % 10)` were replaced by `(10 - (total % 10)) % 10`?

Section 4.2

- W 9. Show the output produced by each of the following program fragments. Assume that i , j , and k are `int` variables.
- (a)

```
i = 7; j = 8;
i *= j + 1;
printf("%d %d", i, j);
```
 - (b)

```
i = j = k = 1;
i += j += k;
printf("%d %d %d", i, j, k);
```
 - (c)

```
i = 1; j = 2; k = 3;
i -= j -= k;
printf("%d %d %d", i, j, k);
```
 - (d)

```
i = 2; j = 1; k = 0;
i *= j *= k;
printf("%d %d %d", i, j, k);
```