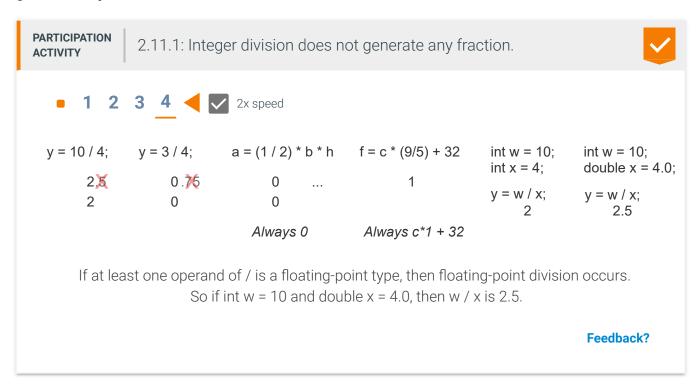
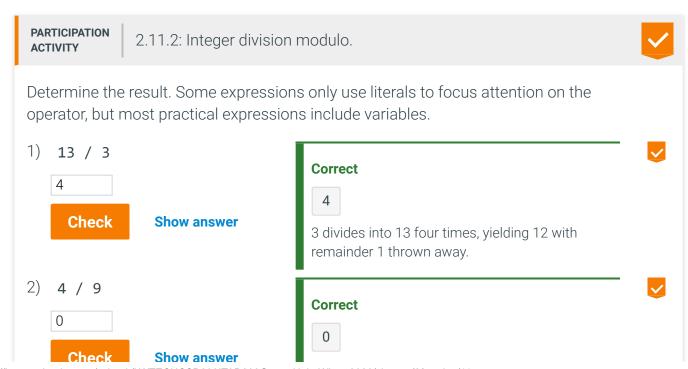
2.11 Integer division and modulo

Division: Integer rounding

When the operands of / are integers, the operator performs integer division, which does not generate any fraction.



The / operator performs floating-point division if at least one operand is a floating-point type.



Division: Divide by 0

Check

For integer division, the second operand of / or % must never be 0, because division by 0 is mathematically undefined. A *divide-by-zero error* occurs at runtime if a divisor is 0, causing a program to terminate. A divide-by-zero error is an example of a *runtime error*, a severe error that occurs at runtime and causes a program to terminate early. In the example below, the program terminates and outputs the error message "Floating point exception" when the program attempts to divide by daysPerYear, which is 0.

4.0 is 2.5.

point type, so floating-point division is applied: 10 /

Figure 2.11.1: Divide-by-zero example: Compute salary per day.

Show answer

Feedback?

```
#include <iostream>
using namespace std;
int main() {
   int salaryPerYear; // User input: Yearly salary
   int daysPerYear; // User input: Days worked per year
   int salaryPerDay; // Output:
                                    Salary per day
   cout << "Enter yearly salary: ";</pre>
   cin >> salaryPerYear;
   cout << "Enter days worked per year: ";</pre>
   cin >> daysPerYear;
   // If daysPerYear is 0, then divide-by-zero causes program
termination.
   salaryPerDay = salaryPerYear / daysPerYear;
   cout << "Salary per day is: " << salaryPerDay << endl;</pre>
   return 0;
}
```

Enter yearly salary: 60000 Enter days worked per year: 0 Floating point exception

Feedback?

PARTICIPATION ACTIVITY

3) 100 * 1 / 2

Check

50

2.11.3: More integer division.



Determine the result. Type "Error" if the program would terminate due to divide-by-zero. Only literals appear in these expressions to focus attention on the operators; most practical expressions include variables.

1) 100 / 2
50
Check Show answer
2) 100 * (1 / 2)
0
Check Show answer

Correct



100 / 2 is just 50.

Correct



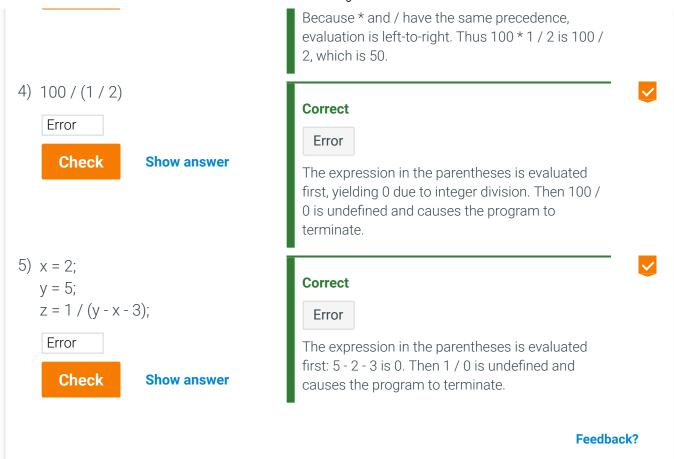
The expression in the parentheses is evaluated first, yielding 0 due to integer division. Then 100 * 0 is 0.

Correct

50

Show answer

2.11. Integer division and modulo



Modulo (%)

The basic arithmetic operators include not just +, -, *, /, but also %. The **modulo operator** (%) evaluates the remainder of the division of two integer operands. Ex: 23 % 10 is 3.

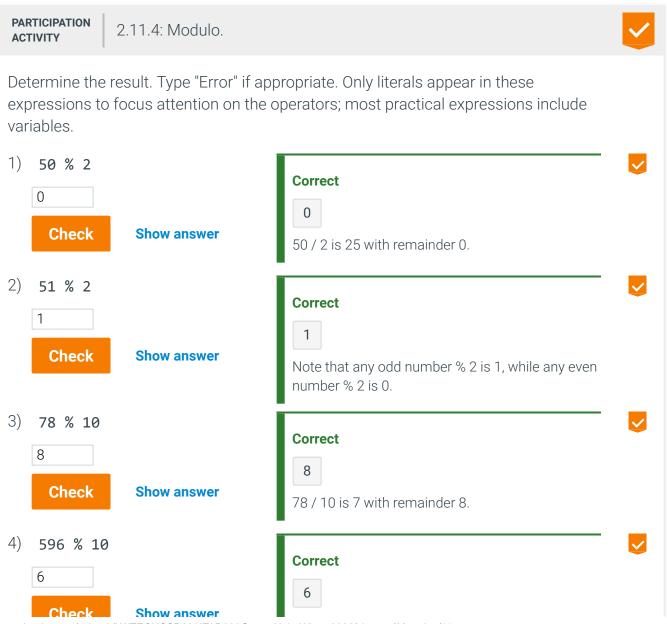
Examples:

- 24 % 10 is 4. Reason: 24 / 10 is 2 with remainder 4.
- 50 % 50 is 0. Reason: 50 / 50 is 1 with remainder 0.
- 1 % 2 is 1. Reason: 1 / 2 is 0 with remainder 1.
- 10 % 4.0 is not valid. "Remainder" only makes sense for integer operands.

Figure 2.11.2: Division and modulo example: Minutes to hours/minutes.

Enter minutes: 367
367 minutes is 6 hours and 7 minutes.
...
Enter minutes: 180
180 minutes is 3 hours and 0 minutes.

Feedback?



2.11. Integer division and modulo

Note that any number % 10 yields the digit in the rightmost (1s) place, in this case 6.

Correct

Error

1/2 is 0. 100 % 0 is undefined (as 100 / 0 is undefined) and causes the program to terminate.

Correct

Error

% is only defined for integer operands; "remainder" makes no sense if floating-point division is performed. 100.0 / 40 is 2.5; no remainder exists because a fraction was generated. In contrast, 100 % 40 is 20 (100 / 40 is 2 remainder 20).

Feedback?

PARTICIPATION ACTIVITY

100 % (1 / 2)

Show answer

Show answer

Error

Error

Check

100.0 % 40

Check

2.11.5: Integer division and modulo.



A florist wants to create as many bunches of 12 flowers as possible. totalFlowers holds the total number of flowers available.

1) Complete the statement to assign numBunches with the maximum number of bunches that can be made.

numBunches = totalFlowers / 12

Check

Show answer

Correct

totalFlowers / 12

totalFlowers / 12 calculates the number of bunches that can be made with exactly 12 flowers. If totalFlowers is 45, 45 / 12 or 3 bunches can be made.

Correct

totalFlowers % 12

totalFlowers % 12 calculates the remainder of totalFlowers divided by 12, which is the number of remaining flowers. If totalFlowers is 45, 3 bunches of 12 can be made, with 45 % 12 or 9 remaining flowers.

 Using only the variable totalFlowers, complete the statement to assign remainingFlowers with the number of remaining flowers after creating as many bunches

of 12 as possible.

```
remainingFlowers =
totalFlowers % 12
  Check
              Show answer
```

Feedback?

Why parentheses matter

The following summary of a dialog on a popular programmer discussion forum shows the importance of using parentheses, in this case in an expression involving modulo.



Use these

(Poster A): Tried rand() % (35 - 18) + 18, but it's wrong.

(Poster B): I don't understand what you're doing with (35 - 18) + 18.

Wouldn't that just be 35?

(Poster C): The % operator has higher precedence than the +

operator. So read that as (rand() % (35 - 18)) + 18.

CHALLENGE ACTIVITY

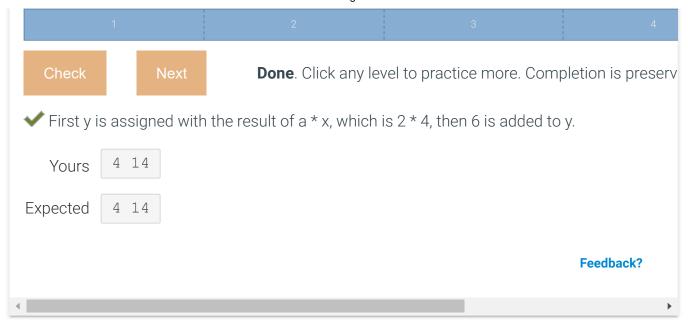
2.11.1: Enter the output of the integer expressions.



Jump to level 1

Type the program's output.

```
#include <iostream>
using namespace std;
int main() {
  int a;
  int x;
  int y;
   a = 2;
   x = 4;
   y = a * x;
   y = y + 6;
   cout << x << " " << y;
   return 0;
```



Modulo examples

Modulo has several useful applications. Below are just a few.

Example 2.11.1: Random number in range.

Given a random number randNum, % can generate a random number within a range:

- randNum % 10
 - Yields 0 9: Possible remainders are 0, 1, ..., 8, 9. Remainder 10 is not possible: Ex: 19 % 10 is 9, but 20 % 10 is 0.
- randNum % 51
 - Yields 0 50: Note that % 50 would yield 0 49.
- (randNum % 9) + 1
 - Yields 1 9: The % 9 yields 9 possible values 0 8, so the + 1 yields 1 9.
- (randNum % 11) + 20
 - Yields 20 30: The % 11 yields 11 possible values 0 10, so the + 20 yields 20 30.

Feedback?

Example 2.11.2: Getting digits.

Given a number, % and / can be used to get each digit. For a 3-digit number userVal like 927:

Feedback?

Example 2.11.3: Get prefix of a phone number.

Given a 10-digit phone number stored as an integer, % and / can be used to get any part, such as the prefix. For phoneNum = 9365551212 (whose prefix is 555):

```
tmpVal = phoneNum / 10000; // / 10000 shifts right by 4, so 936555. prefixNum = tmpVal \% 1000; // \% 1000 gets the right 3 digits, so 555.
```

Dividing by a power of 10 shifts a value right. 321 / 10 is 32. 321 / 100 is 3.

% by a power of 10 gets the rightmost digits. 321 % 10 is 1. 321 % 100 is 21.

Feedback?

PARTICIPATION ACTIVITY

2.11.6: Modulo examples.



- Given a non-negative number x, which yields a number in the range 5 -10?
 - O x % 5
 - O x % 10
 - O x % 11
 - (x % 6) + 5
- 2) Given a non-negative number x, which expression has the range -10 to 10?
 - O x % -10

Correct

% 6 yields 0 - 5. Then + 5 yields 5 - 10.

Correct

x % 21 yields 0 to 20. Then - 10 yields -10 to 10.



(x % 21) - 10

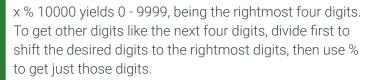
O (x % 20) - 10

- 3) Which gets the tens digit of x. Ex: If x = 693, which yields 9?
 - O x % 10
 - O x % 100
 - (x / 10) % 10
- 4) Given a 16-digit credit card number stored in x, which gets the last (rightmost) four digits? (Assume the fourth digit from the right is nonzero).
 - O x / 10000
 - x % 10000

Correct

x / 10 shifts right one place, putting the tens digit in the ones place. Then % 10 gets the (new) ones digit. Ex: 693 / 10 is 69, then 69 % 10 is 9.

Correct



Feedback?

CHALLENGE ACTIVITY

2.11.2: Compute change.



A cashier distributes change using the maximum number of five dollar bills, followed by one dollar bills. For example, 19 yields 3 fives and 4 ones. Write a single statement that assigns the number of 1 dollar bills to variable numOnes, given amountToChange. Hint: Use the % operator.

```
1 #include <iostream>
2 using namespace std;
4 int main() {
      int amountToChange;
      int numFives;
      int numOnes;
8
      cin >> amountToChange;
9
10
      numFives = amountToChange / 5;
11
      /* Your solution goes here */
12
      numOnes = amountToChange%5;
13
14
      cout << "numFives: " << numFives << endl;</pre>
15
      cout << "numOnes: " << numOnes << endl;</pre>
16
```

