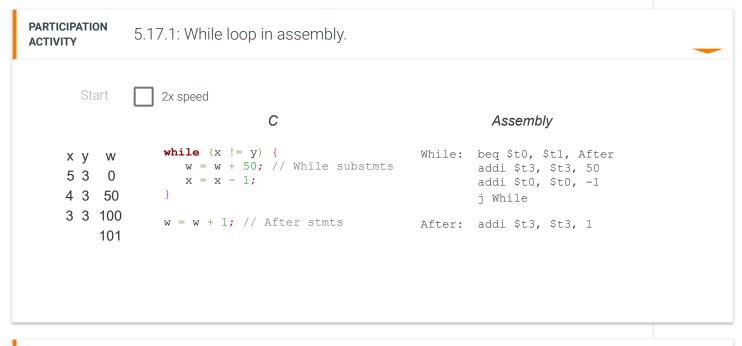
## 5.17 Loops

## While loops

In C, a **while loop** has an expression and substatements. If the expression is true, the substatements execute, and then exe back to check the expression again. Each execution of a loop's substatements is called an **iteration**.

A while loop can be converted to assembly using a pattern similar to an if statement's pattern, but with a jump back after the substatements.



**PARTICIPATION** ACTIVITY 5.17.2: While loop in assembly.

Implement the C by completing the assembly. Assume \$t0 has x's value, \$t1 has y's value, and \$t2 has 2.

```
while (x \le y) {
                                             While: (a) ___ $t0, $t1, After
                                                     (b) ___ $t0, $t0, $t2
(c) ____
                         x = x * 2;
                                             (d)___: addi $t1, $t1, 3
                      y = y + 3;
1) (a)
      Check
                   Show answer
2) (b)
      Check
                   Show answer
3) (c)
      Check
                   Show answer
4) (d)
      Check
                   Show answer
```

## For loops

In C, a **for loop** has four parts: substatements, and three preceding parts of an initialization, an expression, and an update. It merely a convenient representation of a common form of while loop. Thus, to implement in assembly, one can convert the while loop, and then implement the while loop as above.

Figure 5.17.1: For loop first converted to while loop, then to assembly.

```
// for (Init; Expr; Update)
                                i = 0;
                                                //
                                                           addi $t0, $zero, 0 # i = 0
for (i = 0; i < y; i = i +
                                                           While: bge $t0, $t1, After # while (i
                                Init
                                while (i < y) { //
1) {
  w = w + 50; // Substmts
                                Expr
                                                                 addi $t3, $t3, 50 #
                                   W = W + 50; //
                                                           + 50
                                 Substmts
                                                                 addi $t0, $t0, 1 # i = i
                                   i = i + 1; //
                                                           + 1
                                                                 j While
                                Update
                                                           After:
```

In the assembly above, assume \$t0 is i, \$t1 is y, and \$t3 is w.

## PARTICIPATION ACTIVITY

5.17.3: For loop as a while loop.

Arrange the statements to implement the for loop using a while loop.

```
for (i = 50; i >= 0; i = i - 1) {
    x = x + y;
    y = y + 2;
}
```

(1)

(2)

(3)

(4)Reset **CHALLENGE** 5.17.1: Loops in assembly. **ACTIVITY** Start Convert the C to assembly. Variables: w is in \$t0, x is in \$t1, and y is in \$t2. while  $(x \ge y)$  { w = w + 50;x = x - 1;w = w + 10;Registers While: While ▼  $\blacksquare$ 11 \$t0 addi ▼ \$t0 ▼ , \$t0 ▼ , 50 \$t1 9 5 \$t2 addi ▼ \$t0 ▼ \$t0 ▼ After ▼  $\overline{\mathbb{V}}$ After: addi ▼ \$t0 ▼ , \$t0 ▼ , 10 2 3 4 1 Check Next Provide feedback on this section