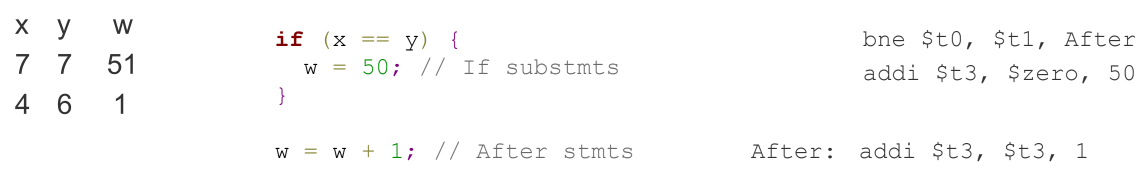
5.15 If-else

If statement

In C, an ***if*** statement executes substatements when the statement's expression is true, otherwise the substatements are sk
expression is a comparison for equality, an if statement can be converted to a simple pattern of assembly instructions star
(branch on not equal) instruction.

PARTICIPATION ACTIVITY 5.15.1: If statement in assembly.

Start □ 2x speed



x	y	w
7	7	51
4	6	1

```
if (x == y) {
  w = 50; // If substmts
}

w = w + 1; // After stmts
```

```
        bne $t0, $t1, After
        addi $t3, $zero, 50

After:  addi $t3, $t3, 1
```

PARTICIPATION ACTIVITY 5.15.2: If statement in assembly.

Variable values: x is $t0, y is $t1, and z is $t2. w's value should be in $t3. For the given C, select the correct assembly.

1) `if (x == y) {`

```
    w = 0;
}

w = w + 5;
```

○
```
         beq  $t0, $t1,
After
         addi $t3, $zero, 0
After:   addi $t3, $t3, 5
```

○
```
         bne $t0, $t1, After
         addi $t3, $zero, 0
After:   addi $t3, $t3, 5
```

2)
```
if (x == y) {
   w = z + 1;
   w = w + y;
}

w = w + 5;
```

○
```
         beq $t0, $t1, After
         addi $t3, $t2, 1
         add $t3, $t3, $t1
After:   addi $t3, $t3, 5
```

○
```
         bne $t0, $t1, After
         addi $t3, $t2, 1
After:   addi $t3, $t3, 5
```

○
```
         bne $t0, $t1, After
         addi $t3, $t2, 1
         add $t3, $t3, $t1
After:   addi $t3, $t3, 5
```

3)
```
if (x == 0) {
   w = w + 10;
}

w = w + 5;
```

○
```
          beq $t0, $zero,
    After
          addi $t3, $t3, 10
    After: addi $t3, $t3, 5
```

○
```
          bne $t0, $zero,
    After
          addi $t3, $t3, 10
    After: addi $t3, $t3, 5
```

If-else statement

In C, an ***if-else*** statement executes one of two possible sets of substatements depending on an expression's value. When t
a comparison for equality, an if-else statement can be converted to a simple pattern in assembly, starting with a bne instru

PARTICIPATION ACTIVITY 5.15.3: If-else statement in assembly.

Start □ 2x speed

x	y	w
2	2	51
0	0	51
4	3	91

```
if (x == y) {                          bne $t0, $t1, Else
   w = 50;                             addi $t3, $zero, 50
}                                      j After
else {
   w = 90;                    Else:    addi $t3, $zero, 90
}

w = w + 1;                    After:   addi $t3, $t3, 1
```

PARTICIPATION ACTIVITY

5.15.4: If-else statement in assembly.

The first few questions list assembly instructions intended to implement the if-else statement, in sequence. Indicate whether the instructions are correct. Assume x and y values are in $t0 and $t1, and w's value should be in $t3.

```
if (x == y) {
   w = 0;
}
else {
   w = x;
}

w = w + 5;
```

1) `(1) bne $t0, $t1, After`

○ Correct

○ Incorrect

2) `(2) addi $t3, $zero, 0  # w = 0;`

○ Correct

○ Incorrect

3) `(3) Else: add $t3, $t0, $zero  # w = x;`

○ Correct

○ Incorrect

4) `(5) After: addi $t3, $t3, 5`

○ Correct

○ Incorrect

5) Suppose a programmer inserted the

instruction `j After` after the Else substatement of `add $t3, $t0, $zero`. Is that jump instruction correct or incorrect?

- ○ Correct
- ○ Incorrect

6) Using the assembly pattern introduced above, does any assembly instruction branch to an If label?

- ○ Yes
- ○ No

If-else-if

C programs commonly use a multi-branch form of an if-else statement. The assembly language pattern is similar to above more labels. Each part's check has a branch (bne) to the subsequent part. Each part (except the last) ends with a jump to A

Figure 5.15.1: If-else-if in assembly.

```
if (x == y) {
   w = w + 50;
}
else if (x == z) { // Else1
   w = w + 60;
}
else {             // Else2
   w = w + 70;
}
```

```
#
        bne $t0, $t1, Else1  # (x == y)
        addi $t3, $t3, 50    # w = w + 50;
        j After

Else1:  bne $t0, $t2, Else2  # (x == z)
        addi $t3, $t3, 60    # w = w + 60;
        j After

Else2:  addi $t3, $t3, 70    # w = w + 70;

After:
```

PARTICIPATION ACTIVITY

5.15.5: If-else-if in assembly.

Find the error in the assembly, which is supposed to implement the if-else-if statement.

```
if (x == y) {
   w = w + 50;
}
else if (x == z) {
   w = w + 60;
}
else {
   w = w + 70;
}
```

1)

```
        bne $t0, $t1, After
        addi $t3, $t3, 50
        j After

Else1:  bne $t0, $t2, Else2
        addi $t3, $t3, 60
        j After

Else2:  addi $t3, $t3, 70

After:
```

PARTICIPATION ACTIVITY

5.15.6: If-else-if in assembly.

Complete the missing assembly instructions to implement the C if-else-if code. $t0 has x's value, $t1 has y's, $t2 has z's. $t3 should get w's value.

```
if (x == z) {
   w = w + 50;
}
else if (x == y) {
   w = w + 60;
}
else {
   w = w + 70;
}
```

```
#
       (A) _____ $t0, $t2, Else1
        addi $t3, $t3, 50
        j After

(B) _____
        addi $t3, $t3, 60    # w = w + 60
        (C) ______

Else2:  addi $t3, $t3, 70    # w = w + 70

After:
```

1) (A)

□ `$t0, $t2, Else1`

Check Show answer

2) (B)

□

Check Show answer

3) (C)

□

Check Show answer

CHALLENGE ACTIVITY 5.15.1: If-else in assembly.

Start

1

Convert the C to assembly. Variables: x is in $t1, y is in $t2, and z is in $t3.

2

```
if (x == z) {
    y = 50;
}
y = y + 1;
```

3

4

5

```
        bne ▼    $t2 ▼,  $t2 ▼,  After ▼
        addi ▼   $t2 ▼,  $t2 ▼,  1
After:  addi ▼   $t2 ▼,  $t2 ▼,  1
```

Registers

$zero	0
$t1	2
$t2	11
$t3	2

1	2	3	4	5

Check Next

Provide feedback on this section