CornellX: ENGRI1210x The Computing Technology Inside Your Smartphone

KarenWest ▼

Courseware **Course Info** **Discussion** **Wiki** **Progress** **Discussion Guidelines** **Resources** **Exploring Engineering**

Syllabus **How to Use Jade**

Help

CONTROL STRUCTURES: IF

0:00 / 4:31 **1.0x**

Download transcript .txt

Show Discussion New Post

1. CHECK YOUR UNDERSTANDING (1/1 point)

Compare the following two pieces of C code:

```
z = 0;               z = 0;
if (x = y)           if (x == y)
  z = x++;             z = x++;
```

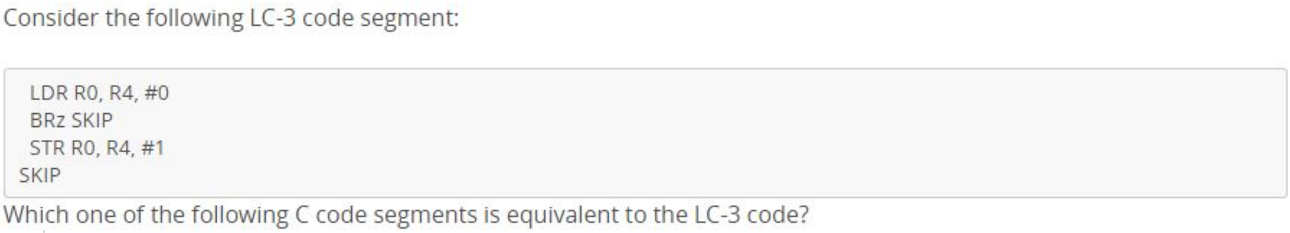
If `x` = 1 and `y` = 2, what would be the value of `z` after executing the code on the left versus on the right?

- ○ Left: 0, Right: 0
- ○ Left: 1, Right: 1
- ● Left: 2, Right: 0 ✓
- ○ Left: 3, Right: 1

Show Answer *You have used 2 of 2 submissions*

Show Discussion New Post

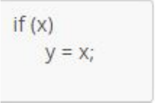
2. CHECK YOUR UNDERSTANDING (1/1 point)



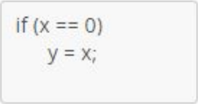
Consider the following LC-3 code segment:

```
 LDR R0, R4, #0
 BRz SKIP
 STR R0, R4, #1
SKIP
```

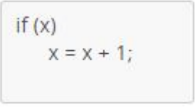
Which one of the following C code segments is equivalent to the LC-3 code?

●  ✓

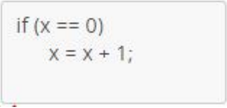
```
if (x)
    y = x;
```

○ 

```
if (x == 0)
    y = x;
```

○ 

```
if (x)
    x = x + 1;
```

○ 

```
if (x == 0)
    x = x + 1;
```

Show Answer *You have used 2 of 2 submissions*

Show Discussion

New Post

Help

CONTROL STRUCTURES: IF-ELSE

0:00 / 3:17

1.0x

Download transcript

.txt

Show Discussion

New Post

At 3:20 and 5:31, the highlighted line should be ADD R0, R0, R2.

CONTROL STRUCTURES: FOR

Help

0:00 / 6:06 1.0x

Download transcript .txt

Show Discussion

New Post

3. CHECK YOUR UNDERSTANDING (1/1 point)

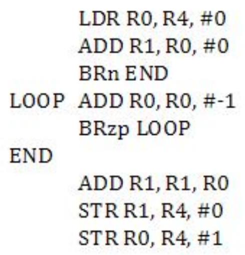
Consider the following C code segment:

```
for (i = n; i >= 0; i--);
    n = n + i;
```

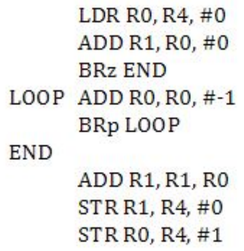
Notice that the programmer accidently put a semicolon after the `for` statement. That means that the next statement `(n = n + i)` is not part of the loop body, and will only execute after the `for` loop completes.

Which of the following LC-3 assembly language programs implements this C segment?

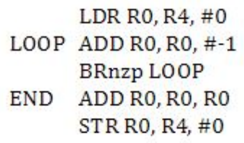
Help

●  ✓

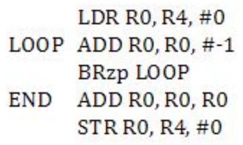
```
      LDR R0, R4, #0
      ADD R1, R0, #0
      BRn END
LOOP  ADD R0, R0, #-1
      BRzp LOOP
END
      ADD R1, R1, R0
      STR R1, R4, #0
      STR R0, R4, #1
```

○ 

```
      LDR R0, R4, #0
      ADD R1, R0, #0
      BRz END
LOOP  ADD R0, R0, #-1
      BRp LOOP
END
      ADD R1, R1, R0
      STR R1, R4, #0
      STR R0, R4, #1
```

○ 

```
      LDR R0, R4, #0
LOOP  ADD R0, R0, #-1
      BRnzp LOOP
END   ADD R0, R0, R0
      STR R0, R4, #0
```

○ 

```
      LDR R0, R4, #0
LOOP  ADD R0, R0, #-1
      BRzp LOOP
END   ADD R0, R0, R0
      STR R0, R4, #0
```

EXPLANATION

The first code is correct. The LDR loads the value of n from memory into R0, which will be used for `i`. Since `n` needs to be retained for the last C statement, the ADD instruction transfers the data from R0 into R1. If that transfer results in the `n` bit being set, then `n` < 0 and the loop is skipped entirely. Otherwise, the next ADD instruction decrements `i`, and the BRzp instruction will cause another iteration to be executed so long as `i` >= 0. The first ADD after the loop performs the statement `n = n + i` using the value of `n` in R1 and the value of `i` in R0. The next two STR instructions store the updated values of n and i.

The second code segment has the wrong nzp designations for the BR instructions.

The last two code segments fail to retain `n` and instead perform `i = i + i` after the loop and overwrite `n` with `i` .

Final Check | **Save** | Hide Answer | *You have used 1 of 2 submissions*

Show Discussion

New Post

Help

EdX offers interactive online classes and MOOCs from the world's best universities. Online courses from MITx, HarvardX, BerkeleyX, UTx and many other universities. Topics include biology, business, chemistry, computer science, economics, finance, electronics, engineering, food and nutrition, history, humanities, law, literature, math, medicine, music, philosophy, physics, science, statistics and more. EdX is a non-profit online initiative created by founding partners Harvard and MIT.

Terms of Service and Honor Code

Privacy Policy (Revised 10/22/2014)

About edX

- About
- News
- Contact
- FAQ
- edX Blog
- Donate to edX
- Jobs at edX

Follow Us

- Facebook
- Twitter
- LinkedIn
- Google+
- Tumblr
- Meetup
- Reddit
- Youtube