

Day 4 (Mon 1/31)

Exercise 3-A & 3-B review

Day 4 recap Qs

Logical operators & control flow

Exercise 4

slido.com
jhuintprog01
↑
zero

Homework 0

due Friday 2/4 by 11 pm

Exercise 3-A

- clone your private github repo

tip: if you are authenticating using a personal access token:

`git config --global credential.helper cache`

- all of your work on exercises & individual homeworks will be here
- don't forget to add/commit/push regularly!
- submission workflow (zip, scp/pscp to download zipfile)

Exercise 3-B

- clone the course's public repo

- starter code will be made available here (don't forget to

⇒ `git pull` regularly)

- copying start code, editing, compiling

Day 4 recap questions

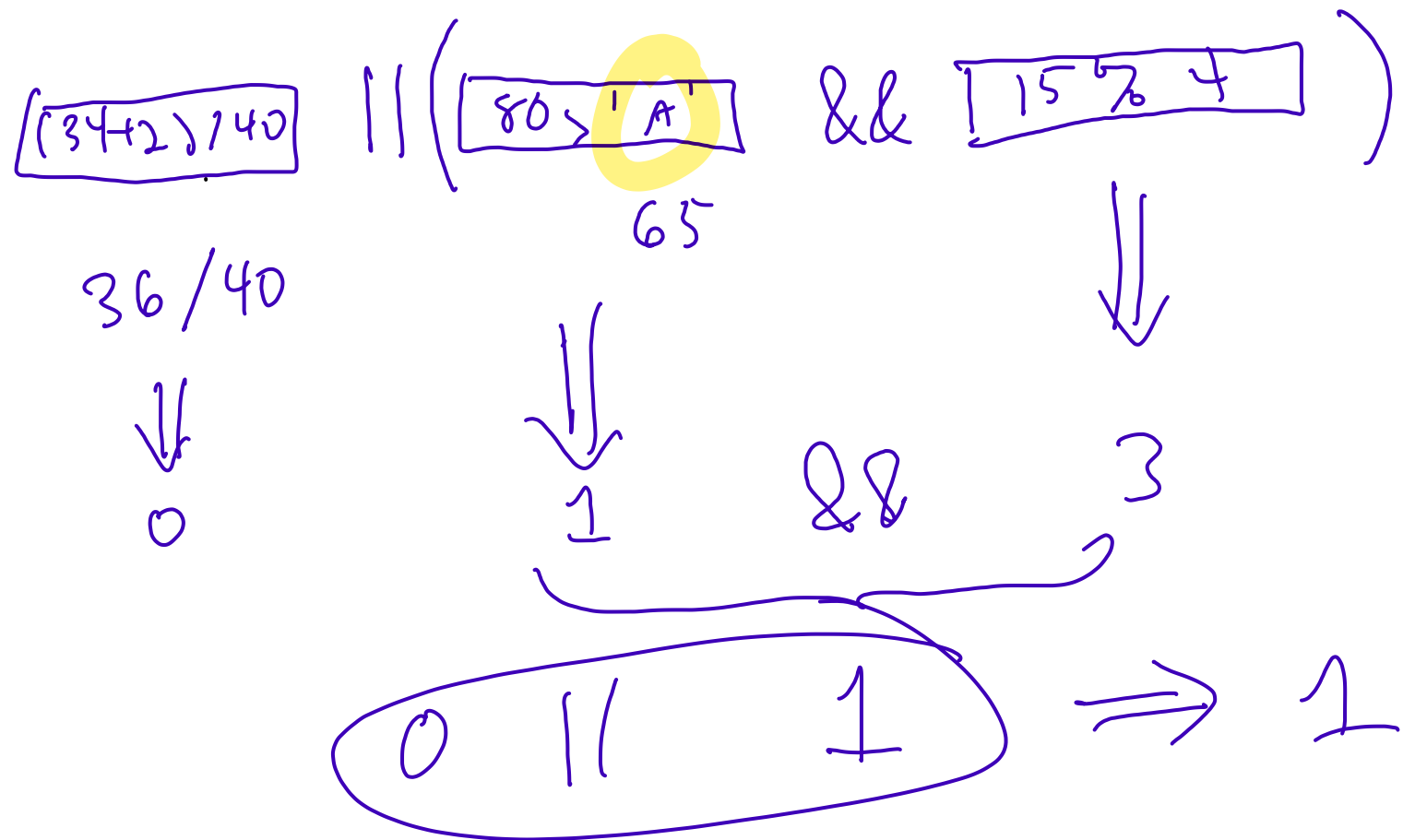
1. Which one is the logical "and" operator in C, && or & or both? *only*
2. Which one is the logical "negation" operator in C, ~ or ! or both? *only*
3. What is the result of evaluating $(34 + 2) / 40 \parallel 80 > 'A' \&\& 15 \% 4$?
4. What does the keyword break do in a control structure?
5. What does the keyword continue do in loops?
6. How many times is the initialize statement in a for loop executed?

terrible code
use parentheses

0 = false

not 0 = true

$\square \parallel \square$



```
while ( cond ) {  
    if ( cond2 ) { break; }  
    if ( cond3 ) { continue; }  
}
```

The diagram illustrates the execution flow of a while loop. A green arrow labeled "avoid" starts from the `break;` statement and points back to the `cond` condition, indicating that the loop is exited. An orange arrow starts from the `continue;` statement and points down to a crossed-out asterisk, indicating that the current iteration is skipped and the loop continues from the start.

"early
return"

```
for ( A ; B ; C ) {  
    body  
}
```

⇓ equivalent

```
A  
while ( B ) {  
    body  
    C  
}
```