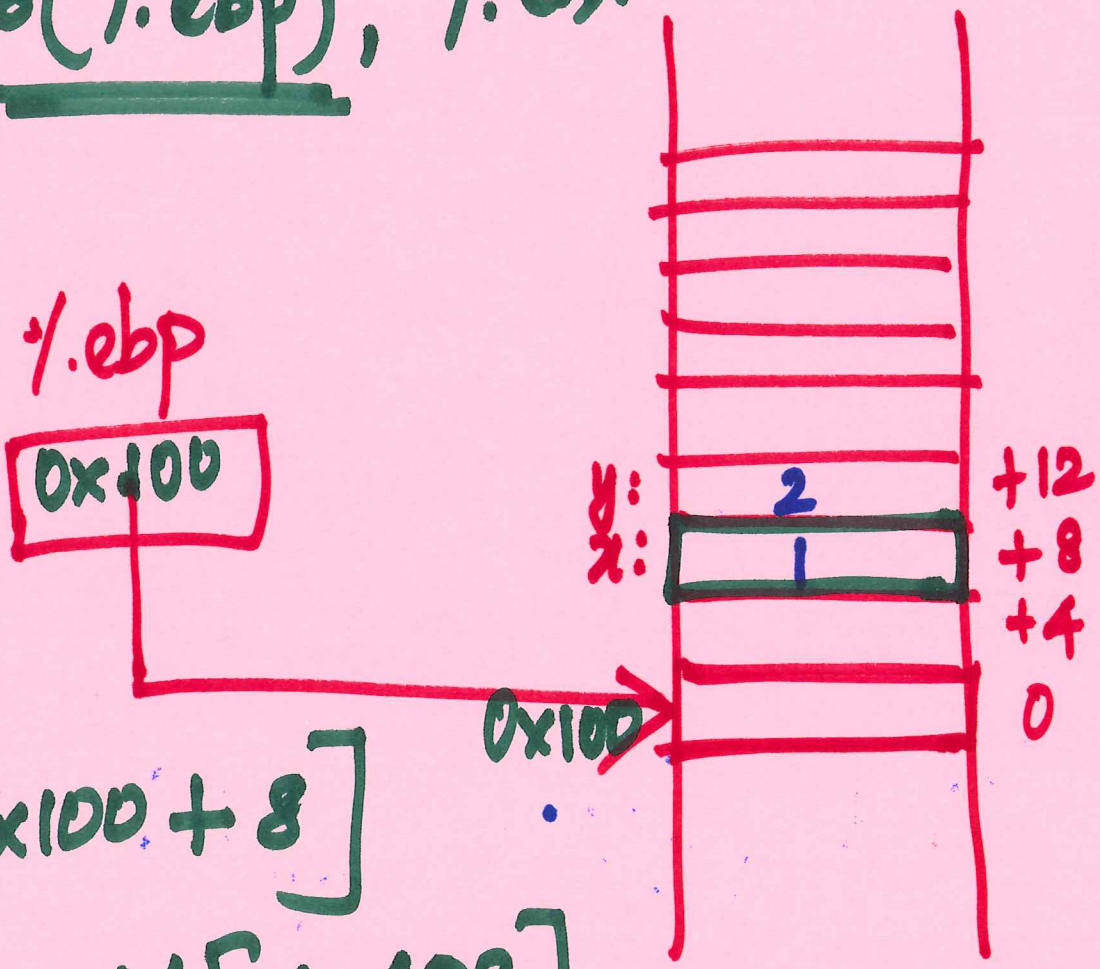
Lecture - 16

movl 8(%.ebp), %.eax



M[0x100 + 8]

= M[0x108]

%.eax 1

movl $12(%ebp), %eax

subl 8(%ebp), %eax

%eax: 2

subl s, d

d ← d − s

%eax ← 2 − 1

%eax: 1

Return value

int or ptr

↳ always in %eax

addl (s), (%eax)

↓

register or imm.

cmpl 12(%ebp), %eax

y=2

cmpl s_2, s_1

$s_1 - s_2$

a - b

int → signed 2's comp.

1-2 = -1

ZF 0

SF 1

OF 0

jge .L2

jl

jge

SF ∧ OF

~(SF ∧ OF)

~(1 ∧ 0) = ~1 = 0

9: jge 08

b:

b → 11

8

19

0x13

0x 80483F7

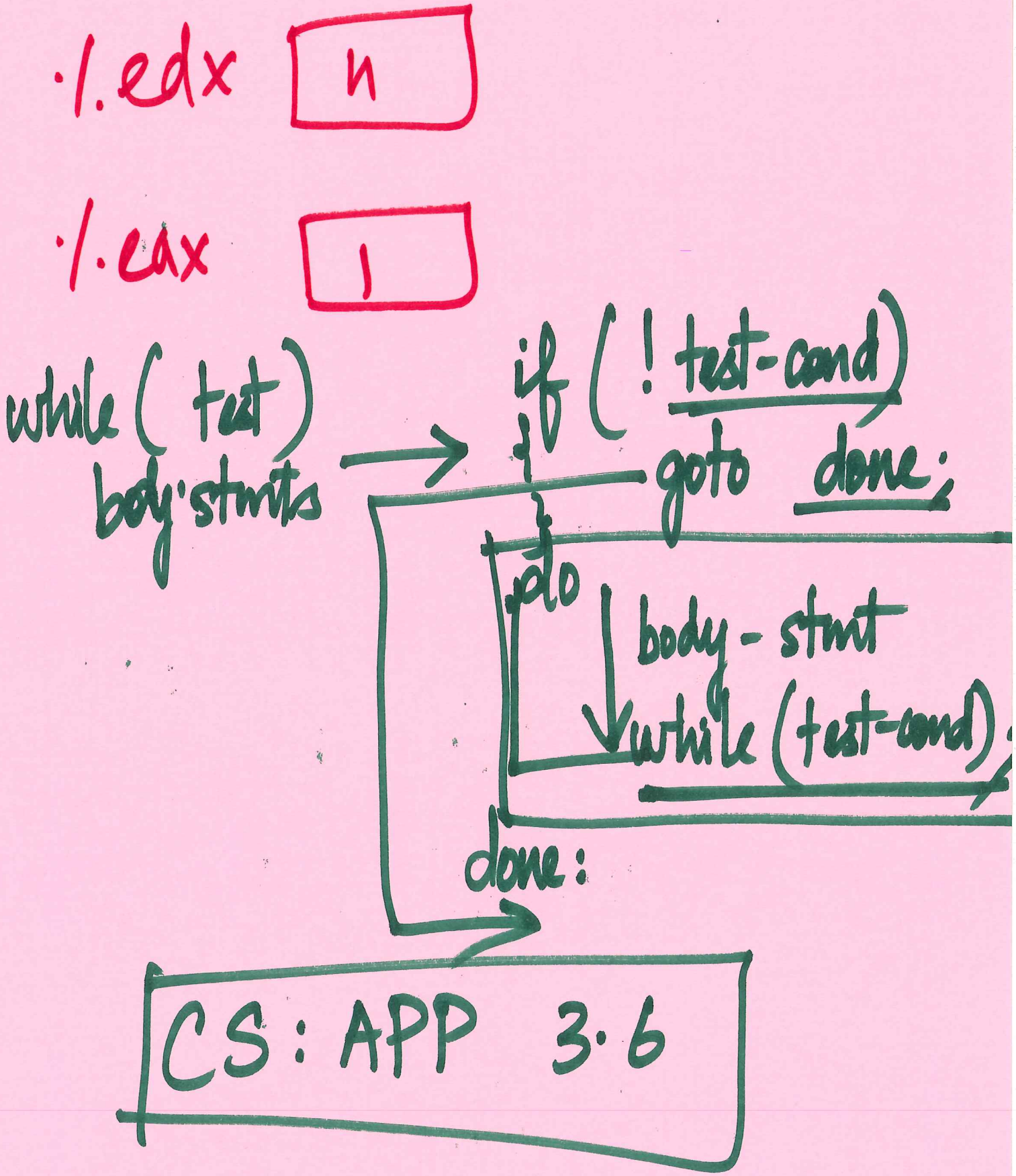
08

0x 80483FF

for , while, do-while

```
do {
   ==========
   ==========
   ==========
} while (cond);
```

```
loop:
        body-stmts
        t = test-cond;
        if (t)
            goto loop;
```


%edx
n
%eax
1
while (test)
body-stmts
if (! test-cond)
goto done;
loop
body-stmt
while (test-cond)
done:
CS:APP 3.6