

Computation Structures 2: Computer Architecture

<u>Help</u>

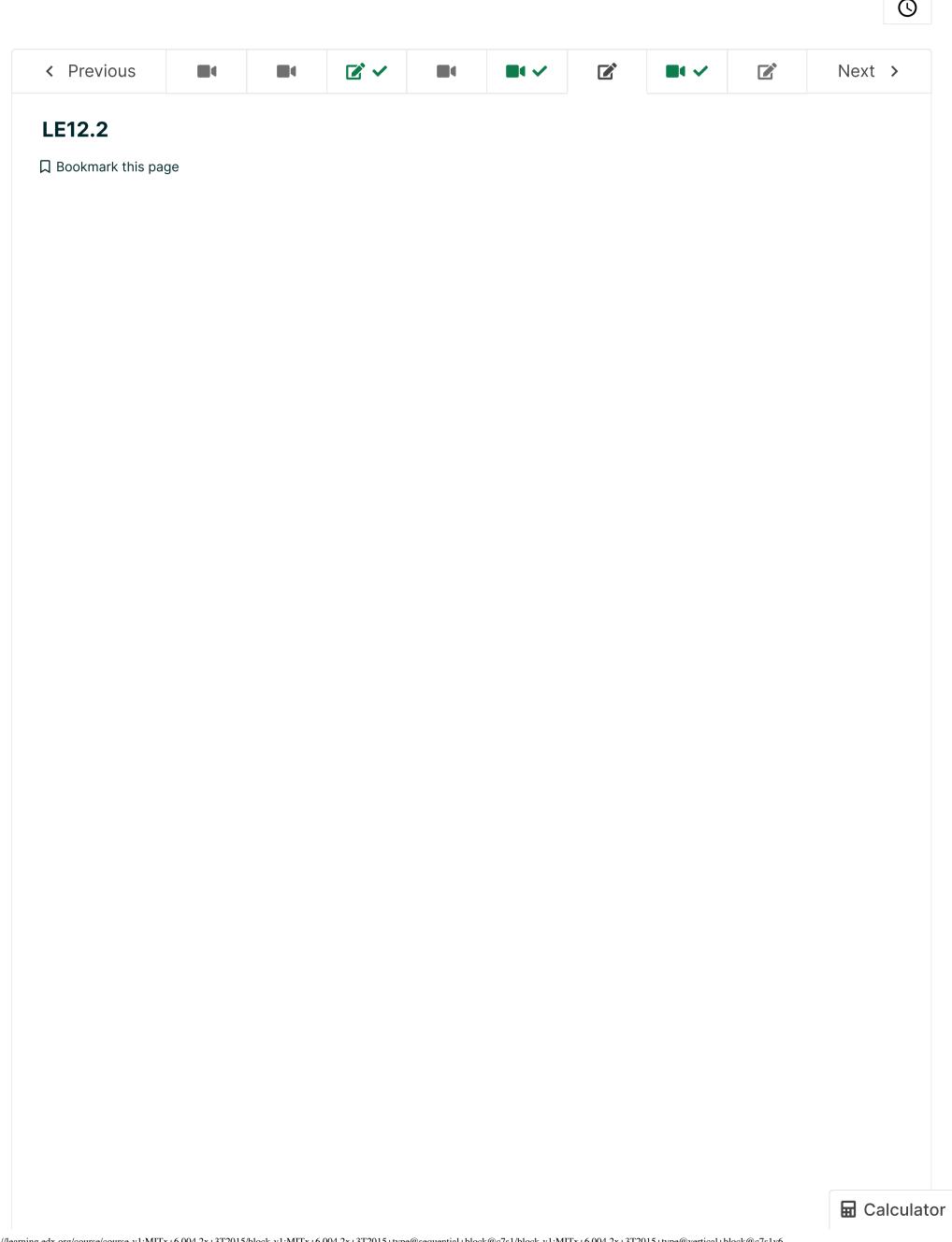




Discussion <u>Course</u> <u>Progress</u> <u>Dates</u>

☆ Course / 12. Procedures and Stacks / Lecture Videos (35:16)





LE12.2.1 Compiling a Procedure

3/4 points (ungraded)

You are given the following listing of a C program and its translation to Beta assembly code:

```
int ones(int x) {
  int lowbit = x \& 1;
                         // low bit of x
  int rest = x \gg 1;
                         // shift other bits right
  if (x == 0) return 0;
  else return ones(rest)+lowbit
}
```

```
[01]
     ones:
             PUSH(LP)
[02]
             PUSH(BP)
[03]
             MOVE(SP, BP)
[04]
             ALLOCATE(2)
[05]
             PUSH(R1)
[06]
             LD(BP, -12, R0)
[07]
             ANDC(R0, 1, R1)
[08]
             ST(R1, 0, BP)
[09]
             SHRC(R0, 1, R1)
             ST(R1, 4, BP)
[10]
[11]
             BEQ(R0, labl)
[12]
             LD(BP, 4, R1)
[13]
             PUSH(R1)
             BR(ones, LP)
[14]
[15]
             DEALLOCATE(1)
[16]
             LD(BP, 0, R1)
[17]
             ADD(R0, R1, R0)
            POP(R1)
[18]
     labl:
[19]
             MOVE(BP, SP)
[20]
             POP(BP)
[21]
             POP(LP)
[22]
             JMP(LP)
```

(A) Which line(s) of assembly language form the entry sequence? Include code responsible for allocating local

variables and saving the values of registers used in the body of the procedure. Check all the applicable lines.
[01] ones:PUSH(LP)
✓ [02] PUSH(BP)
✓ [03] MOVE(SP, BP)
[04] ALLOCATE(2)
[05] PUSH(R1)
[06] LD(BP, -12, R0)
[07] ANDC(R0, 1, R1)
[08] ST(R1, 0, BP)
[09] SHRC(R0, 1, R1)
[10] ST(R1, 4, BP)
[11] BEQ(R0, lab1)
[12] LD(BP, 4, R1)

[13] PUSH(R1)

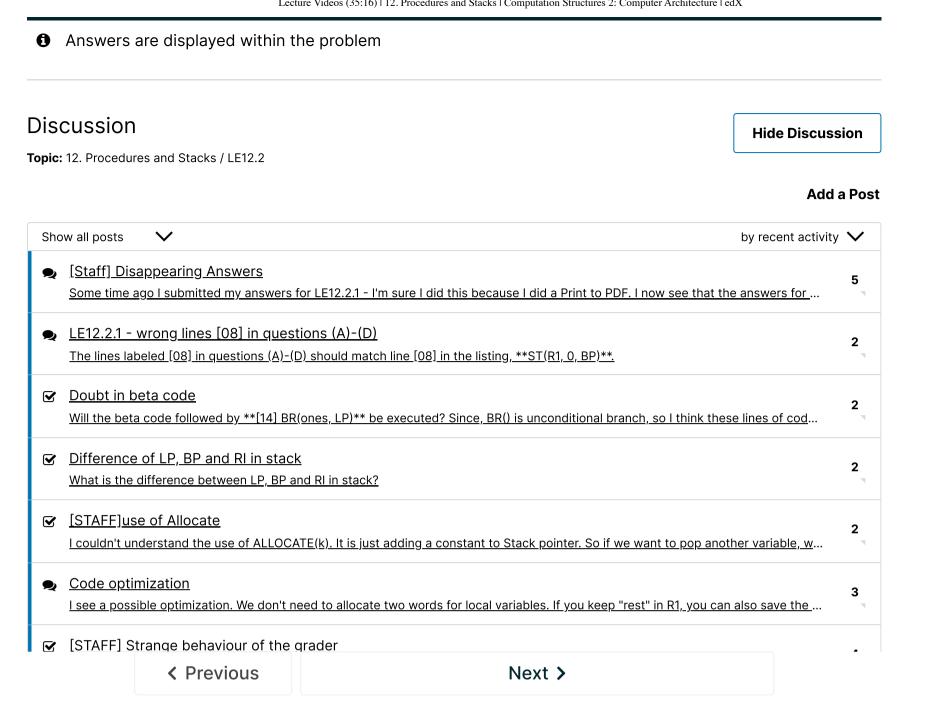
⊞ Calculator

[14] E	BR(ones, LP)
[15] [DEALLOCATE(1)
[16] [_D(BP, 0, R1)
[17]	ADD(R0, R1, R0)
[18]	labl:POP(R1)
[19] M	10VE(BP, SP)
[20] F	POP(BP)
[21] F	P0P(LP)
[22] 3	JMP(LP)
	ne(s) of assembly language form the exit sequence? Include code responsible restoring values saved sequence. Check all the applicable lines.
[01]	ones:PUSH(LP)
[02] F	PUSH(BP)
[03] M	MOVE(SP, BP)
[04] A	ALLOCATE(2)
[05] F	PUSH(R1)
[06] L	_D(BP, −12, R0)
[07] A	ANDC(RØ, 1, R1)
[08]	ST(R1, 0, BP)
[09] 5	SHRC(RØ, 1, R1)
[10] 9	ST(R1, 4, BP)
[11] E	BEQ(R0, lab1)
[12] [_D(BP, 4, R1)
[13] F	PUSH(R1)
[14] E	BR(ones, LP)
[15] [DEALLOCATE(1)
[16] L	_D(BP, 0, R1)

[18] labl:POP(R1) [19] MOVE(BP, SP) [20] POP(BP) √ [21] POP(LP) [22] JMP(LP) ~ (C) Which line(s) of assembly language were generated when compiling the recursive call ones(rest). Check all the applicable lines. [01] ones:PUSH(LP) (02) PUSH(BP) ✓ [03] MOVE(SP, BP) [04] ALLOCATE(2) [05] PUSH(R1) ✓ [06] LD(BP, -12, R0) [07] ANDC(R0, 1, R1) [08] ST(R1, 0, BP) [09] SHRC(R0, 1, R1) [10] ST(R1, 4, BP) [11] BEQ(R0, lab1) √ [12] LD(BP, 4, R1) [13] PUSH(R1) [14] BR(ones, LP) [15] DEALLOCATE(1) [16] LD(BP, 0, R1) [17] ADD(R0, R1, R0) [18] labl:POP(R1) [19] MOVE(BP, SP) [20] POP(BP) **⊞** Calculator

	JMP(LP)
	line(s) of assembly language were generated when compiling the statement "lowbit = $x \& 1$;" Check able lines.
[01]	ones:PUSH(LP)
[02]	PUSH(BP)
[03]	MOVE(SP, BP)
[04]	ALLOCATE(2)
[05]	PUSH(R1)
[06]	LD(BP, -12, R0)
(07]	ANDC(R0, 1, R1)
[08]	ST(R1, 0, BP)
[09]	SHRC(R0, 1, R1)
[10]	ST(R1, 4, BP)
[11]	BEQ(R0, lab1)
[12]	LD(BP, 4, R1)
[13]	PUSH(R1)
[14]	BR(ones, LP)
[15]	DEALLOCATE(1)
[16]	LD(BP, 0, R1)
[17]	ADD(R0, R1, R0)
[18]	labl:POP(R1)
[19]	MOVE(BP, SP)
[20]	POP(BP)
[21]	POP(LP)
[22]	JMP(LP)
~	

■ Calculator



© All Rights Reserved



edX

<u>About</u>

<u>Affiliates</u>

edX for Business

Open edX

Careers

<u>News</u>

Legal

Terms of Service & Honor Code

Privacy Policy

Accessibility Policy

Trademark Policy

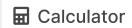
<u>Sitemap</u>

Cookie Policy

Your Privacy Choices

Connect

Idea Hub



Contact Us Help Center

<u>Security</u>

Media Kit















© 2024 edX LLC. All rights reserved.

深圳市恒宇博科技有限公司 <u>粤ICP备17044299号-2</u>