CS 2400 Practice Midterm 2

Solutions

October 14, 2017

1

(a) **20** (+6)

Since %rdi is the first input argument to function p().

(b) **1** (+6)

Since inside the p() function where we call function r(), we only move one value to %rdi right before the call instruction to r(). Thus r() takes only one input argument.

Partial credit: 2 (+3)

(c) **40** (+6)

Number of bytes = 8 (push %rbp) + 24 (sub $0x18, %rsp) + 8 (callq q or for callq r) = 40.

Partial credit: 40 ± 8 (+4.5)

Partial credit: 40 ± 16 (+3)

Partial credit: 40 ± 32 (+1.5)

(d) **120** (+6)

main: 8 (push %rbp) + 16 (sub $0x10, %rsp) + 8 (callq p) p: 8 (push %rbp) + 24 (sub $0x18, %rsp) + 8 (callq q) q: 8 (push %rbp) + 24 (sub $0x18, %rsp) + 8 (callq r) r: 8 (push %rbp)

Thus adding all up we get 120 bytes offset.

Partial credit: 120 ± 10 (+4.5)

Partial credit: 120 ± 20 (+3)

Partial credit: 120 ± 40 (+1.5)

(e) **80** (+6)

Partial credit: 80 ± 8 (+4.5)

Partial credit: 80 ± 16 (+3)

Partial credit: 80 ± 32 (+1.5)

2

Q1:

0x30	0x31	0x32	0x33	0x34	0x35	0x36	0x37	0x38	0x39	0x3a	0x3b	0x3c	0x3d	0x3e	0x3f
76	75	62	6e	65	72	61	6f	62	65	21	61	61	61	61	00 \\
v	u	l	n	e	r	a	b	l	e	!	a	a	a	a	0\\

Grading:
+12 point
+1 for each correct answer

Q2:
8 bytes for pushing return address at (0x40055e)
8 bytes for pushing %rbp at(0x4004d6)
Total bytes = 16
Grading:
+4 for 16
+2 for 8

Q3:
3d: since the bottom of the stack is 0x35 and the pop instruction implicitly
increments the stack pointer (rsp) by 8 (64-bit address)
Grading:
+4 for 3d
+2 for 3c

Q4:
Return address = 0x72616f6265216161 (64-bit address)
representing the characters "rable!aa"
Grading:
+6 for 0x72616f6265216161 or 72616f6265216161 or 0x61612165626f6172 or 61612165626f6172
+3 for 0x72616f62 or 72616f62 or 0x626f6172 or 626f6172

3

Format A Bits	Exponent (E)	Mantissa (M)	Final Value $(M \times 2^E)$	Format B Bits	Final Value
0111 0000	0	1.0	1	01111 000	1
1001 1001	2	1.5625	6.25	10001 100	6

Points break down–

- Format A

  Exponent **+3**, Mantissa **+3**, Final value **+4**.

- Format B

  +8 for bit pattern (1 point per bit), **+4** for correct final value answer but **+2** for final value incorrectly rounded to **5**.