

# CS 2400 Midterm 1

## Solutions

September 20, 2017

### 1

(a) True (b) omitted (c) False (d) True

### 2

Answer key:

1)– c)

2)– f)

3)– d)

4)– e)

### 3

since this is worth 18 points, give +3 to the 'x+2' expression blank, and +3 to last row's overflow blank. Give +2 to other 6 blanks.

y = -29

x = 30

Expression	Decimal Representation	Hex Representation	Overflow?
y	-29	0x23	No
<b>x + 1</b>	TMax	<b>0x1f</b>	No
<b>x + 2</b>	-32	0x20	Yes
x+y	1	0x01	<b>No</b>
x + TMax	<b>-3</b>	0x3d	Yes
TMin+y	<b>3</b>	<b>0x03</b>	<b>Yes</b>

### 4

omitted

## 5

```

push    %ebp
mov     __ESP__,%ebp
sub     $0x34,%esp
mov     0x8(%ebp),%eax
mov     %a1,-0x34(%ebp)
movl    $0xf,-0x20(%ebp)
mov     0xc(__EBP__),%eax
mov     %ax,-0x22(%ebp)
__JMP__    L1

.L4
__CMPL__    $0x0,-0x20(%ebp)
jns      L2
movzwl    -0x22(%ebp),%eax
sub       $0x1,%eax
movsbl    -0x34(%ebp),__ECX__
movzwl    -0x22(%ebp),%edx
__ADD__    %ecx,%edx
mov       %edx,-0x1c(%ebp,__EAX__,4)
jmp       L3

.L2
__MOVZWL__ -0x22(%ebp),%eax          // note: some put movl here, but this is wrong
because it would move the lower 2 bytes from int y into the upper 2 bytes of eax
movsbl    -0x34(%ebp),%edx
mov       __EDX__,-0x1c(%ebp,%eax,4) // note: some answers put 0x8(%ebp) here,
which is incorrect because an x86 assembly instruction cannot have both operands
referencing memory

.L3
movzwl    -0x22(%ebp),%eax
mov       -0x1c(%ebp,%eax,4),%eax
add       %eax,-0x20(%ebp)
movzwl    -0x22(%ebp),%eax
__SUB__    $0x1,%eax
mov       %ax,-0x22(%ebp)

.L1
cmpw     $0x0,-0x22(%ebp)
jne      .L4
mov       -0x20(%ebp),%eax
leave
ret

```