Exam 1 Solutions
15-213 / 18-213 Fall 2011

*********
Problem 1
*********
1-c 2-b 3-c 4-c 5-d 6-c 7-b 8-c

*********
Problem 2
*********
umax: 63
tmin: -32
(unsigned) ((int) 4): 4
(unsigned) ((int) -7): 57
((unsigned) 0x21) << 1) & 0x3f: 0x02
(int) (20 + 12): -32
12 && 4: 1
(! 0x15) > 16: 0

*********
Problem 3
*********
| Value | FP bits | Rounded value | |
|---|---|---|---|
| 1 | 100 10 | 3 | normalized, exact |
| 9 | 110 00 | 8 | normalized, round down, because 10 is odd |
| 3/16 | 000 11 | 3/16 | exact, denorm |
| 15/2 | 110 00 | 8 | normalized, round up, change exponent |

*********
Problem 4
*********
1st blank: n == 0 or n < 1
2nd blank: n >> 1 or n / 2

*********
Problem 5
*********
Part A:

| | |
|---|---|
| unknown | 0xffff1008 |
| 18 | 0xffff1004 |
| 213 | 0xffff1000 |
| unknown | 0xffff0ffc |
| unknown | 0xffff0ff8 |
| unknown | 0xffff0ff4 |
| unknown | 0xffff0ff0 |
| 15 | 0xffff0fec |
| 18 | 0xffff0fe8 |
| 0x080483b7 | 0xffff0fe4 |
| 0xffff0ff8 | 0xffff0fe0 |
| unknown | 0xffff0fdc |
| unknown | 0xffff0fd8 |
| 3 | 0xffff0fd4 |
| 15 | 0xffff0fd0 |
| 0x080483b7 | 0xffff0fcc |

```
    +-------------------------------+
    |           0xffff0fe0          |  0xffff0fc8
    +-------------------------------+
    |            unknown            |  0xffff0fc4
    +-------------------------------+
    |            unknown            |  0xffff0fc0
    +-------------------------------+
    |                0              |  0xffff0fbc
    +-------------------------------+
    |                3              |  0xffff0fb8
    +-------------------------------+
    |            unknown            |  0xffff0fb4
    +-------------------------------+
    |            unknown            |  0xffff0fb0
    +-------------------------------+
```

Part B:
esp: 0xffff0fcc
ebp: 0xffff0fe0

Grading Rubric:

Part A (8):
Let m be the number of mistakes.  The following is a
non-exhaustive list of possible mistakes:

- An entry is left blank, although it should contain a value.
- An entry is filled in, although it should remain blank.
- An entry contains an incorrect value.
- The stack is shifted up or down by four bytes.  (If the stack
  is shifted by more than four bytes, count the number of times
  it is shifted.)

In general, 8 - ceil(m / 2) points should be awarded for this
subproblem.  However, at the grader's discretion, one or two
extra points can be awarded for solutions that display some
knowledge of x86 stack conventions, or one or two extra points
can be deducted for solutions that betray only feeble knowledge
of x86 stack conventions.

Part B (2):
One point each for %esp and %ebp.  The addresses must be
consistent with the values provided in the stack diagram.

*********
Problem 6
*********
Part A:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| a | XX | XX | XX | XX | XX | XX | XX |
| b | b | b | b | b | b | b | b |
| c | c | XX | XX | XX | XX | XX | XX |
| d[0] | d[0] | d[0] | d[0] | d[0] | d[0] | d[0] | d[0] |
| d[1] | d[1] | d[1] | d[1] | d[1] | d[1] | d[1] | d[1] |
| e[0] | e[1] | e[2] | XX | f | f | f | f |

end

Part B:

```
(gdb) disassemble foo
Dump of assembler code for function foo:
0x00000000004004e4 <+0>:    sub    $0x8,%rsp
0x00000000004004e8 <+4>:    movb   $0x65,(%rdi)
0x00000000004004eb <+7>:    movq   $0x0,0x18(%rdi)
0x00000000004004f3 <+15>:   movw   $0x213,0x10(%rdi)
0x00000000004004f9 <+21>:   movzbl 0x29(%rdi),%ecx
0x00000000004004fd <+25>:   lea    0x2c(%rdi),%rdx
0x0000000000400501 <+29>:   mov    0x8(%rdi),%rsi
```

```
0x0000000000400505 <+33>:   mov    $0x40062c,%edi
0x000000000040050a <+38>:   mov    $0x0,%eax
0x000000000040050f <+43>:   callq  0x4003e0 <printf@plt>
0x0000000000400514 <+48>:   add    $0x8,%rsp
0x0000000000400518 <+52>:   retq
End of assembler dump.
```

Part C:
Most compact ordering is 40 bytes. for example: b, d, f, e, a, c

*********
Problem 7
*********

```
int test(int x, int y, int z)
{
  int result = 3;
  switch(z)
  {
    case 0:
      x = x & 25;
    case 3:
    case 7:
      result = x;
      break;
    case 5:
      result = 2 * x;
    case 4:
      result = result + y;
      break;
    default:
      result = y;
  }
  return result;
}
```

*********
Problem 8
*********
Answer: (B) 8.
For (A) the sequence of accesses will result in M H M M M M, hit rate = 0.16
For (B) the sequence of accesses will result in M H H M M M, hit rate = 0.33
For (C) the sequence of accesses will result in M H H H M M, hit rate = 0.5
Hence, the answer is (B)

*********
Problem 9
*********

The unstated assumption is that C[i][j] is stored in a register.
However, the question as written is ambiguous because it doesn't state
this assumption clearly. Thus we will accept a miss rate of either 1/2
or 1/4 for part A.

Part A:
16 floats in a block
For row-wise A: Pattern is 1 miss, 15 hits, 1 miss, 15 hits, ...
For col-wise B: Pattern is miss, miss, miss, ...
For C[i][j] not in register: pattern is miss, hit, hit, hit, hit, ...
If C[i][j] in register, then there are zero memory accesses to C[i][j] during
each inner loop.

If you assumed that C[i][j] is in a register, then there are only
accesses to A and B in the inner loop. Thus, the miss rate = 17/32 ~
1/2

If you assumed that C[i][j] is not in a register, then we have to
include the accesses to C in the miss rate calulation: Thus, the miss
rate = 17/48, which is closer to 1/4 than 1/2.

Part B:
For row-wise A, B: Pattern is 1 miss, 15 hits, 1 miss, 15 hits, ...

If you assumed that C[i][j] is in a register, then the miss rate is 2/32 = 1/16.

If you assumed that C[i][j] is not in a register, then the miss rate is 2/48 = 1/24, and the closest rate on the answer key is still 1/16.