Exam 1 Solutions
15-213 / 18-213 Fall 2012

****
Problem 1
****
1-a 2-c 3-d 4-c 5-a 6-b 7-c 8-(b or d) 9-c 10-d

The correct answer for 8 was initially listed as d) temporal locality,
but the correct answer is actually spatial locality. While it's true
that blocking in things like matmult primarily exploits temporal
locality, blocking is effective for transpose because it exploits
spatial locality by effectively using the entries in each cache line;
there is no reuse.

****
Problem 2
****

Expression	4b decimal	4b binary	6b decimal	6b binary
-8	-8	1000	-8	11 1000
-TMin	-8	1000	-32	10 0000
x >> 1	-3	1101	-3	11 1101
(-x ^ -1) >> 2	-2	1110	-2	11 1110

****
Problem 3
****

	A	B	
One	0 011 00	0 01 000	Exact in both formats
1/2	0 010 00	0 00 100	Exact in both formats, norm in A, denorm in B
11/8	0 011 10	0 01 011	Format A round to even, format B exact

****
Problem 4
****
```
unsigned transform(unsigned n)
{
    int b, m;

    for(m = 0; n != 0; n >>= 1) { // (or) for(m = 0; n > 0; n = n/2)
        b = n & 1; // (or) b = n % 2;

        if(b == 0) {
            continue;
        }

        m = 2*m + 1; // (or) m = m + m + 1; (or) m = m<<1 + 1;
    }

    return m;
}
```

Alternate solution:

```
unsigned transform(unsigned n)
{
    int b, m;

    for(m = 0; n != 0;) {
        b = !(n & 1); // (or) b = (n % 2) - 1;
```

```
        if(b == 0) {
            m = 2*m + 1;
        }

        n = n >> 1;
    }

    return m;
}
```

```
*********
Problem 5
*********
Part 1.
a X X X X X X b b b b b b b b
c c c c d d d X e e e e e e e e
f f f f f f f

Part 2.
f f f f f f f b b b b b b b b
e e e e e e e c c c c d d d a

or

a d d d c c c c b b b b b b b b
e e e e e e e e f f f f f f f f
```

```
*********
Problem 6
*********
A: phd
B: bachelors
C: masters
```

```
*********
Problem 7
*********
int result = 4;

switch(a){
    case 0:
    case 1:
        c = c - 5;
    case 2:
        result = 4 * c; //or result *= c
        break;
    case 5:
        result = 86547; //or 0x15213
        break;
    case 3:
        c = 2;
    case 7:
        b = b & c;
    default:
        result += b; // or result = b + 4
}

return result;
}
```

```
*********
Problem 8
*********
Stack      The diagram starts with the
addresss   arguments for foo()
```

```
          +-----------------------------------+
0xffffd850|                5                  |
          +-----------------------------------+
0xffffd84c|                4                  |
          +-----------------------------------+
0xffffd848|                3                  |
          +-----------------------------------+
0xffffd844| caller ra: 0x080483c9             |
          +-----------------------------------+
0xffffd840| old ebp: ffffd858                 | <- Part B: %ebp=0xffffd840
          +-----------------------------------+
0xffffd83c|                3                  |
          +-----------------------------------+
0xffffd838|                4                  |
          +-----------------------------------+
0xffffd834| foo ra: 0x08048397                | <- Part C: esp=0xffffd834
          +-----------------------------------+
0xffffd830| old ebp: 0xffffd840               | ok to omit, not part of the stack
anymore
          +-----------------------------------+
0xffffd82c|                                   |
          +-----------------------------------+
0xffffd824|                                   |
          +-----------------------------------+
}

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

A. TTSSSBBB

B.
Set:Tag:hit/miss
0:1:M
6:2:M
0:1:H
7:3:M
6:2:H
2:2:M
2:3:M
6:2:H
4:1:M
0:0:M

C. Final state: 0 X 3 X 1 X 2 3 (c)
```