EECS 213
Introduction to Computer Systems
Midterm Exam

1. (16 pts total) Given the C code on the right:

   a) (6 pts) `gcc -S` produces the assembly code below. Explain what each line does.

   ```c
   int bitcnt(int n)
   {
     unsigned m = 0;
     while ( n > 0 ) {
       m += n & 0x1;
     }
     return m;
   }
   ```

   ```assembly
   pushl %ebp
   movl %esp, %ebp
   subl $16, %esp
   movl $0, -4(%ebp)
   jmp L2
   L3:
     movl 8(%ebp), %eax
     andl $1, %eax
     addl %eax, -4(%ebp)
   L2:
     cmpl $0, 8(%ebp)
     jg L3
     movl -4(%ebp), %eax
     leave
     ret
   ```
b) (6 pts) gcc -S -O2 produces this assembly code. Explain what each line does.

```
pushl %ebp ____________________________
movl %esp, %ebp ____________________________
movl 8(%ebp), %eax ____________________________
testl %eax, %eax ____________________________
jg L5 ____________________________
xorl %eax, %eax ____________________________
pl %ebp ____________________________
ret ____________________________
L5:
jmp L5 ____________________________
```

c) (4 pts) Explain the optimizations made in version (b).
2. (6 pts) `strlen()` in C returns the length of a string. Its prototype is:

```c
typedef unsigned int size_t;
size_t strlen(const char * s);
```

A student who didn’t take EECS 213 wrote this code:

```c
int is_longer_str(const char *s1, const char *s2)
{
    return strlen(s1) - strlen(s2) > 0;
}
```

Give an example where this will do the wrong thing, explain why, and give a simple fix. Be specific.

3. (13 pts) Fill in the following table for an IEEE floating point representation with 1 sign bit S, 3 exponent bits and 3 fraction bits, M should be an integer or fraction, e.g., 0, 1, \( \frac{3}{4} \). M, E and V should be base 10. \( V = (-1)^{S} \times M \times 2^{E} \)

<table>
<thead>
<tr>
<th>Binary</th>
<th>M</th>
<th>E</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 000 000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 110 110</td>
<td>0.75</td>
<td>1.75</td>
<td>1.75</td>
</tr>
<tr>
<td>0 000 011</td>
<td></td>
<td></td>
<td>∞</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4. (19 pts) Fill in the table for a 5-bit two’s complement integer representation.

<table>
<thead>
<tr>
<th>Name</th>
<th>Decimal</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>14</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-9</td>
<td>0 1100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 0100</td>
</tr>
<tr>
<td>Tmax</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tmin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tmin + Tmax</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tmin + 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tmax + 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-TMax</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-TMin</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5. (15 pts) Given:

```c
typedef struct {
    char c;
    double p;
    float d;
    short s;
    int *i;
} Struct1;
```

A. Use vertical lines and labels to indicate clearly how data would be allocated for each element of a structure of type `Struct1` on an IA32 (x86) machine using Linux alignment rules. Crosshatch areas that are allocated but not used.

B. How many bytes are allocated for an object of type `Struct1`?

C. What alignment is required for an object of type `Struct1`? I.e., if an object must be aligned on an $x$-byte boundary, then say what $x$ is.

D. Do (A) again, with the fields of `Struct1` re-ordered to use the least number of bytes. Crosshatch areas that are allocated but not used.
6. (14 pts) Assume the variables $a$ and $b$ are signed integers. Assume two’s complement representation. Assume that $\text{MAX_INT}$ is the maximum integer, $\text{MIN_INT}$ is the minimum integer, and $W$ is word length minus one, e.g., $W = 31$ for 32-bit integers. Next to each item on the left, write the letter of the code on the right that best matches it.

<table>
<thead>
<tr>
<th>Description</th>
<th>Choice</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td></td>
<td>a. $\neg(a \mid (b \oplus (\text{MIN_INT} + \text{MAX_INT})))$</td>
</tr>
<tr>
<td>$a &amp; b$</td>
<td></td>
<td>b. $((a \oplus b) &amp; \neg b) \mid (\neg(a \oplus b) &amp; b)$</td>
</tr>
<tr>
<td>$a \ast 7$</td>
<td></td>
<td>c. $a \gg 3$</td>
</tr>
<tr>
<td>$a / 8$</td>
<td></td>
<td>d. $\neg((a \gg W) \ll 1)$</td>
</tr>
<tr>
<td>$(a &lt; 0) \ ? 1 : -1$</td>
<td></td>
<td>e. $((a &lt; 0) \ ? (a + 7) : a) \gg 3$</td>
</tr>
<tr>
<td>$a \ast 14$</td>
<td></td>
<td>f. $((\neg a &amp; b) \mid a) &amp; ((\neg a &amp; b) \mid \neg b)$</td>
</tr>
<tr>
<td>$a \oplus b$</td>
<td></td>
<td>g. $\neg((a \mid (\neg a + 1)) \gg W) &amp; 1$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>h. $(a \ll 3) + (a \ll 2) + (a \ll 1)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>i. $1 + (a \ll 3) + \neg a$</td>
</tr>
</tbody>
</table>