CS61C, Fall 1997
Midterm #1

Problem #1 (3 points)

Convert the eight-bit binary value 11110000 to:
(a) hexadecimal.
(b) decimal, interpreting it as a unsigned value.
(c) decimal, interpreting it as a twos complement signed value.

Problem #2 (3 points)

Decode the following binary numbers as MIPS instructions and give the equivalent MIPS assembly language (MAL) statements.

```
address   value
0x40      10001100101101110000000000100100
0x44      00000010111001001011000000100011
0x48      00011110110000001111111111110000
```

Problem #3 (2 points)

Why did the MIPS designers use PC-relative branch addressing (One sentance is enough!)

Problem #4 (4 points)

Consider this C struct definition:

```
struct foo {
   int *p;
   int a[3];
   struct foo *sf;
} baz;
```

Suppose that register $16 contains the address of baz.
For each of the following C statements, indicate which of the MAL code fragments below (A-H) could be the result of compiling it.

```
codeA: lw    $8, 0($16)
       sw    $8, 4($16)

codeB: lw    $8, 0($16)
       lw    $9, 0($8)
       sw    $9, 4($16)

codeC: lw    $8, 4($16)
       sw    $8, 0($16)

codeD: sw    $16, 16($16)

codeE: lw    $17, 6($16)

codeF: lw    $17, 12($16)

codeG: lw    $8, 0($16)
       sw    $8, 16($16)

codeH: addi  $8, $16, 4
       sw    $8, 0($16)
```

____ number = baz.a[2];

____ baz.p = baz.a;

____ baz.a[0] = *baz.p;

____ baz.sf = &baz;

Problem #5 (6 points)

Translate the following C procedure to MAL. Use the convention in which arguments are passed in registers.

```
int garply(int a, int *b) {
    int c;

    c = subt(a >> 6);
    *b = a + *b;
    if (a < ) || c < 0)
       return c;
    else
       return c | a;
}
```

Problem #6 (6 points)

Consider the following fragment of a C/C++ program.

```
int v[10], s;
int *p;

s = 17;
for (p = &v[3]; *p != 0; p++)
        s = s + *p;
```

Here is a buggy translation in MAL, assuming s is in $16 and p is in $19.

```
          or    $16, $0, $0
          lw    $19, v+12
loop:
          bne   $8, finish
          add   $16,$19,$16
          addi  $19, 1
```

```
          j     loop
finish:
```

There are six errors, including one missing instruction, in this translation. Find and fix them.
