1. Show the machine code (in binary and hex) for the following MIPS code segment.

```
lw $s0, 2($s7)
and $t0, $s0, 0x8
srl $t0, $t0, 3
```

For the remaining problems, translate the C code segments into MIPS assembly.

2. Assume that `int a, b, c, and d` are located in registers `$s0, $s1, $s2, and $s3`, respectively.

```
if (a == b)
    d = c;
else if (a == c)
    d = b;
else
    d = a;
```

3. Assume that `int n, i, and sum` are located in registers `$s0, $s1, and $s2`, respectively.

```
sum = 0;
for (i=1; i < n; i++)
    sum += i;
```

4. Assume that the address of array `int A[]` is located in register `$s7` and that `int n, i, and sum` are located in registers `$s0, $s1, and $s2`.

```
sum = 0;
for (i=0; i < n; i++)
    sum += A[i];
```

5. Translate the following C function into MIPS assembly.

```
int arraysum(int array[], int size) {
    int i, sum;
    sum = 0;
    for (i=0; i < size; i++)
        sum += array[i];
    return sum;
}
```

6. Translate the following C function call into assembly. Assume `$s5` contains the address of the array `int numbers[]`, `$s6` contains the size `int n`, and the `int` value `total` will be stored in `$s7`.

```
total = arraysum(numbers, n);
```
7. Translate the following recursive C function into MIPS assembly.

```c
int terminal(int n) {
    if (n < 1)
        return 0;
    else
        return (n + terminal(n - 1));
}
```

Acknowledgments  This derivative work of Janet Davis and Marge Coahran, used under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License was developed in collaboration with Charlie Curtsinger.