```
/*
 * hw6.c
 *
 *  Created on: Oct 24, 2014
 *      Author: Scot Matson
 *          HW: 06
 *      Course: CS469C
 *     Section: 01
 * Description: Binary manipulation
 */

#include <stdio.h>
#define BITS_PER_BYTE 8

/* Print a list of 1-bit positions */
// Modified functions from Howell midterm study gui
de.
int bitList(unsigned int n) {
    int bitCount = 0;
    int i = -1 + sizeof(int) * BITS_PER_BYTE;
    for (; i>=0; i--) {
        if (n & 1<<i) {
            ++bitCount;
        }
    }
    return (bitCount % 2 == 0) ? 0 : 1;
}

int bits(int n, unsigned int start, unsigned int pa
ttern, char *fn) {
    FILE *fptr = fopen(fn, "wt");
    int b;
    int mask, period = 0;
    unsigned int x;

    unsigned int patt1 = pattern >> 1; //Shifting r
ight, this is done once.
    x = start;
    do {
        mask = patt1 & x;
        b = bitList(mask);
        fprintf(fptr, "%d ", b);
```

```
        x = x << 1;
        x = x + b;
        x = x << (32 - n);
        x = x >> (32 - n);
        ++period;
    } while (x != start);

    fprintf(fptr, "\n");
    fclose(fptr);
    return period;
}
```