```
#
# Makefile for spiralworm
# Author: Michael Kepple
# Date:   28 Mar 2013
#
spiralworm: spiralworm.h spiralworm.c
        gcc spiralworm.c spiralworm.h -lncurses -o spiralworm
/*
 * spiralworm.c
 *
 * Author: Michael Kepple
 * Date:   25 Mar 2013
 * Description: 'worm-like' text slides around the borders of the screen.
*/
#include "spiralworm.h"

/*
 * Function: enqueue
 * Description: for passed in queue, adds a new element at the rear of the queue.
 * Params: Queue to add to, x & y coordinates to store in new Node.
 * Returns: Nothing.
 * Modifies: queue
*/
void enqueue(Queue * queue, int y, int x, char ch)
{
    Node * new = calloc(1, sizeof(Node));
    new->x = x;
    new->y = y;
    new->ch = ch;
    if (queue->head == NULL && queue->tail == NULL)
    {
        queue->size++;
        queue->head = queue->tail = new;
    }
    else if (queue->size < MAXLENGTH)
    {
        queue->size++;
        queue->tail->next = new;
        queue->tail = new;
    }
    else
    {
        free(dequeue(queue));
        enqueue(queue, x, y, ch);
    }
}

/*
 * Function: dequeue
 * Description: dequeues the head element and returns it.
 * Params: the queue to remove from.
 * Returns: the removed element from the queue.
 * Modifies: queue.
*/
Node * dequeue(Queue * queue)
{
    Node * deleteMe = queue->head;
    queue->head = queue->head->next;
    queue->size--;
    return deleteMe;
}

/*
```

```
     * Function: printWorm
     * Description: prints out the current worm representation to the screen.
     * Params: the worm queue.
     * Returns: Nothing.
     * Modifies: Nothing.
    */
    void printWorm(Queue * queue)
    {
        Node * temp = queue->head;
        while (temp != NULL)
        {
            mvwaddch(stdscr, temp->x, temp->y, temp->ch);
            temp = temp->next;
        }
    }

    int main()
    {
        initscr();
        Queue * theQueue = calloc(MAXLENGTH, sizeof(Queue));
        char text[] = "@#$%";
        noecho();
        curs_set(0);
        struct timespec waitTime;
        waitTime.tv_nsec = NANOSLEEP;
        int xPlace, yPlace, iter = 0;
        int currCol = COLS-1;
        int currLines = LINES-1;
        for (iter = 0; iter < MAXLENGTH; iter++)
            enqueue(theQueue, 0, 0, text[0]);
        iter = 0;
        for (;;)
        {
            clear();
            xPlace = 0;
            yPlace = 0;
            while (xPlace < currCol)
            {
                clear();
                xPlace++;
                enqueue(theQueue, iter, (iter+xPlace), text[0]);
                printWorm(theQueue);
                refresh();
                nanosleep(&waitTime, NULL);
            }
            while (yPlace < currLines)
            {
                clear();
                yPlace++;
                enqueue(theQueue, (iter+yPlace), (currCol+iter), text[1]);
                printWorm(theQueue);
                refresh();
                nanosleep(&waitTime, NULL);
            }
            while (xPlace > 0)
            {
                clear();
                xPlace--;
                enqueue(theQueue, (currLines+iter), (iter+xPlace), text[2]);
                printWorm(theQueue);
                refresh();
                nanosleep(&waitTime, NULL);
            }
```

```
            while (yPlace > 1)
            {
                clear();
                yPlace--;
                enqueue(theQueue, (iter+yPlace), iter, text[3]);
                printWorm(theQueue);
                refresh();
                nanosleep(&waitTime, NULL);
            }
            currCol -= 2;
            currLines -= 2;
            if (currCol <= 2 || currLines <= 2)
            {
                flash();
                sleep(1);
                endwin();
                while (theQueue->head != NULL)
                    free(dequeue(theQueue));
                free(theQueue);
                exit(0);
            }
            iter++;
        }
    }

    /*
     * File:   spiralworm.h
     * Author: Michael Kepple
     * Date:   28 Mar 2013
     *
    */
    #ifndef SPIRALWORM_H
    #define SPIRALWORM_H
    #include <stdlib.h>
    #include <ncurses.h>
    #include <time.h>
    #define MAXLENGTH 7
    #define NANOSLEEP 10000000L
    typedef struct Node Node;
    typedef struct Queue Queue;
    struct Node
    {
        int x;
        int y;
        char ch;
        Node * next;
    };
    struct Queue
    {
        Node * head;
        Node * tail;
        int size;
    };
    void enqueue(Queue * queue, int x, int y, char ch);
    Node * dequeue(Queue * queue);
    #endif
```