// Interrupt Exercise (SDCC)

#include <at89c51RC2.h>
#include <stdio.h>

#define BUSY_MASK 0x80
#define WRITE_OPERATION 0x22
#define READ_OPERATION 0x44
#define COMMANDREGISTER ((xdata unsigned char *) 0xF000)
#define DATAREGISTER ((xdata unsigned char *) 0xF001)
#define STATUSREGISTER ((xdata unsigned char *) 0xF002)

void device_write(unsigned char);  //Function prototype
extern void init_interrupts(void);
extern void serinit(int);

unsigned char ii;             //Loop variable
xdata unsigned char *eptr;    //Pointer to external data memory space

main()
{
    serinit(9600);            //Initialize serial to 9600 baud
    EA = 1;                   //Enable interrupts
    init_interrupts();        //Assume this function exists
    ii = 0;                   //Initialize loop variable
    while(1) {                //To write to this device, first you wait until the busy flag is
        device_write(ii++);  //clear. Then send the write command to the command register and
        ii = ii % 254;        //then you send the data value to the data register
    }
}

// Assume Timer 0 interrupt occurs every 20ms
void timer0int(void) __interrupt (1){
    printf("Timer 0 Interrupt\n");
}

// Assume external interrupt 0 occurs every 2 seconds
void ext0int(void) __interrupt (0) {
    device_write(0xff);       //write value 255
}

// Assume external interrupt 1 occurs every 10 seconds
void ext1int(void) __interrupt (2) {
    _asm
        CPL P1.1
        RET
    _endasm;
}