// Interrupt Exercise

#include <8051int.h>
#include <8051reg.h>
#include <8051io.h>

#define BUSY_MASK 0x80
#define WRITE_OPERATION 0x22
#define READ_OPERATION 0x44
#define COMMAND_REGISTER 0xF000
#define DATA_REGISTER 0xF001
#define STATUS_REGISTER 0xF002

void device_write();          //Function prototype

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

main()
{
    serinit(9600);            //Initialize serial to 9600 baud
    enable();                 //Enable interrupts
    init_interrupts();        //Assume this function exists
    ii=0;                     //Initialize loop variable
    while(1)  {
        device_write(ii++);     //Write value, then increment ii
        ii=ii%254;              //ii sequences from 0 to 254
    }
}

// To write to this device, first you wait until the busy flag is
// clear. Then send the write command to the command register and
// then you send the data value to the data register
void device_write(unsigned char value)
{
    eptr=STATUS_REGISTER;
    while((*eptr & BUSY_MASK) != 0); //Wait until busy flag clear
    eptr=COMMAND_REGISTER;
    *eptr=WRITE_OPERATION;    //Identify this as a write operation
    eptr=DATA_REGISTER;
    *eptr=value;              //Write the value
}

// Assume Timer 0 interrupt occurs every 20ms
INTERRUPT(_TF0_) timer0int() {  
    printf("Timer 0 Interrupt\n");
}

// Assume external interrupt 0 occurs every 2 seconds
INTERRUPT(_IE0_) ext0int() {
    device_write(0xff);       //write value 255
}

// Assume external interrupt 1 occurs every 10 seconds
INTERRUPT(_IE1_) ext1int() {
    asm {
        CPL P1.1
        RETI
    }
}