```c
/* ANRC RHKI */
/* Lab8: Interrupt Handler Lab */
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>

#include <linux/interrupt.h>
#include <linux/io.h>

#define DRIVER_AUTHOR "ANRC"
#define DRIVER_DESC   "Lab8"

MODULE_LICENSE("GPL");           // Get rid of taint message by declaring code as GPL.

/*  Or with defines, like this: */
MODULE_AUTHOR(DRIVER_AUTHOR);    // Who wrote this module?
MODULE_DESCRIPTION(DRIVER_DESC); // What does this module do?

int init(void);
void cleanup(void);

/* service keyboard interrupts handler */
irq_handler_t irq_handler(int irq, void *dev_id, struct pt_regs *regs)
{
        static unsigned char scancode;

        /* read keyboard */
        scancode = inb( 0x60 );
        if((scancode == 0x01) || (scancode ==0x81)) printk("interkey: ESC pressed\n");

        return (irq_handler_t) IRQ_HANDLED;
}

/* register the irq handler */
static int keybrd_int_register(void)
{
        int result;
        /* request irq 1 for keyboard using request_irq */

        /* check for success/failure */
        if(result) printk("interkey: failed to get shared interrupt for keyboard irq 1");

        return result;
}

/* remove the handler */
static void keybrd_int_unregister(void)
{
        /* free irq handler using free_irq */
}

int init(void)
{
        printk(KERN_INFO "init_module() called\n");

        printk("interkey: registering keyboard interrupt handler\n");
        keybrd_int_register();

        return 0;
}

void cleanup(void)
{
        printk("interkey: unregistering keyboard interrupt handler\n");
        keybrd_int_unregister();
        printk(KERN_ALERT "Unloading interkey ...\n");
}
```

```
module_init(init);
module_exit(cleanup);
```