```
                         PreLAB3 Work

                        DUE: 2-13-2020
========================================================================
REQUIRED: Show to TA YOUR KBD driver of 1.
          Turn in a hardcopy of your (text-edited) diagram of 2
========================================================================

1. Duplicate and run the KBD driver program C3.2.

   If the keys are INCORRECT, your Ubuntu's QEMU is using scan code set #2.
   You need to use keymap2 to convert scan code into ASCII
   Google scan code set 2 to get/see the scan code of keys.

   Download samples/keymap2 and use it in YOUR kbd driver

For QENU using keyset #2:

   Key press  : ONE interrupt : data = scan code of key
   key release: TWO interrupts, data = 0xF0, followed by data = scan code of key

The KBD driver in C3.2 can only handle lowercase keys.

REQUIRED: Modify it to handle both lowercase and uppercase keys.
          Also: catch Control-C key: print "Contro-C key"
                catch Control-D key: set input char to 0x4 (for EOF)

HINT : you must detect Left-shit/Left-Ctrl pressed but NOT yet released.


2. In an ARM system supporting IRQ interrupts, e.g. KBD interrupts,
   the following components are needed/provided:

(1). Vector tabble at memory address 0
     0x18: LDR PC, irq_handler_addr
     irq_handler_addr: .word irq_handler

(2). irq_handler:
       sub lr, lr, #4
       stmfd sp!, {r0-r12, lr}
       bl  IRQ_handler
       ldmfd sp!, {r0-r12, pc}^

(3). IRQ_handler{
       if (VIC.statusBit31 && SIC.statusBit3)
           kbd_handler();
     }

   int hasData = 0;
   char c;

(4). kbd_handler()
     {
       get scancode;
       c = ASCII char mapped by scancode;
       hasData = 1;
     }

(5). char kgetc()
     {
        while(hasData==0);
        hasData = 0;
        return c;
     }

(6). main()
     {
        unlock();    // allow CPU to accept IRQ interrupts
        kgetc();     // CPU executes this
     }


Assume: the CPU executes kgetc() in main().

1. Draw a diagram to show the control flow of CPU when a KBD key is pressed

                  KCW's BAD Answer Example:
------------------------------------------------------------------------
                           key
In (5) at while(hasData==0); =====> (1) Reason: GOD says so
   (1) =======================> (4) Reason: CPU has a mind of its own

--------------- YOU finish the diagram with valid reasons -----------
```