* Elaborate on your program source code.
  + how the interrupt could be enabled
    - * **The interrupt is enabled with the following code: ori $a0, 0xff11**
  + What happened when the user type a character at the keyboard
    - * **It prints the ascii code for the desired character**
  + how your program notice the interrupt occurs
    - * **The program notices the interrupt when a key is entered so it goes to the kernel to process the code to see what the exception/interrupt is.**
  + how do we figure out the exception type
    - * **The exception type is based on the exception code. In this case it’s I/O which is always 0**
  + Before we return to the user program, what should the kernel program do?
    - * **Restore other registers**
      * **Set status register**
      * **Interrupts get enabled**
      * **Write back to status**
      * **Return to EPC**

The program logic works as follows:

* Enable interrupts
* Keep main in an infinite loop with the jump to itself
* Use kernel segment to process exceptions
* Allow input by user via keyboard
* Get exception code (always 0 for I/O)
* Branch to exit program if q is entered
* If not q, print all ascii codes entered to console
* Continue until user enters a q on the keyboard