```
lab10.s
            Wed Feb 22 17:12:16 2023
// Ezekiel Kim
// Lab 10
// Program to find largest number
// Ezekiel Kim
// /2023
//
        .qlobal _start // Provide program starting address
       .data
               dbX:
                      .quad
               dbY:
                     .quad 0
               szX: .skip 21
               szY: .skip 21
               szMsgl: .asciz "Enter x: "
               szMsg2: .asciz "Enter y: "
               szGt: .asciz " > "
                     .asciz " == "
               szEq:
               szVar1: .asciz "x"
                              "у"
               szVar2: .asciz
               szCol: .asciz ":"
               chCr: .byte 10
       .text
_start:
// Prompt user for X
                     =szMsg1 // String to print
       ldr x0,
       bl
               putstring
                             // Receive x value
       ldr
              x0, =szX
       bl
              getstring
       ldr
              x0, =szX
       bl
              ascint64
              x1, =dbX // Convert to int
       ldr
                     [x1] // Store into dbX
       str
              x0,
// Prompt user for Y
    ldr x0, =szMsg2 // String to print
       bl
              putstring
                             // Receive y value
       ldr
             x0, =szY
       bl
             getstring
       ldr
             x0, =szY
              ascint64
       bl
             x1, =dbY // Convert to int x0, [x1] // Store into dbX
       ldr
       str
// Compute
            x0, =dbX // Load dbX
x0, [x0] // Dereference
x1, =dbY // Load dbY
x1, [x1] // Dereference
       ldr
       ldr
             x1,
x1,
       ldr
                     [x1]
       ldr
             x0,
       cmp
                      x1
       b.eq _equal
       w.it _lessthan
b.gt area*
              _greaterthan
_equal:
             x0,
                     =szVar1 // Print X
       ldr
       bl
              putstring
              x0, =szEq // Print equal sign
       ldr
              putstring
       bl
              x0, =szVar2 // Print Y
       ldr
       bl
              putstring
       ldr
              x0, =szCol // Print semicolon
       bl
              putstring
                              // Print X
       ldr
              x0, =szX
       bl
              putstring
       ldr
              x0, =szEq // Print equal sign
```

bl

ldr

bl

putstring

putstring

x0, =szY

// Print Y

```
lab10.s
            Wed Feb 22 17:12:16 2023
              x0, =chCr // Print carriage return putch
       ldr
       bl
       b.al
              _end
_lessthan:
       ldr x0, =szVar2 // Print Y
bl putstring
ldr x0, =szGt // Print gt sign
       bl
              putstring
       ldr
              x0, =szVar1 // Print X
       bl
              putstring
              x0, =szCol // Print semicolon
       ldr
       bl
              putstring
       ldr x0, =szY // Print Y
       bl
              putstring
       ldr x0, =szGt // Print gt sign
bl putstring
       ldr x0, =szX bl putstring
                             // Print X
       ldr
              x0, =chCr // Print carriage return
              putch
       bl
       b.al
              _end
_greaterthan:
            x0,
                      =szVar1 // Print X
       ldr
              putstring
       bl
               x0, =szGt // Print gt sign
       ldr
              putstring
x0, =szVar2 // Print Y
       bl
       ldr
              putstring
x0, =szCol // Print semicolon
       bl
       ldr
              putstring
       bl
              x0, =szX
                              // Print X
       ldr
       ldr x0, =szX // Print X
bl putstring
ldr x0, =szGt // Print gt sign
bl putstring
       bl
              putstring
       ldr
              x0, =szY
                              // Print Y
       bl
              putstring
       ldr
              x0, =chCr // Print carriage return
              putch
       bl
       b.al _end
// Setup the parameters to exit the program and then call Linux to do it.
_end:
                      #0
               х0,
                               // Sets return code to 0
       mov
                      #0 // Sets return code to 0
#93 // Service command code 93 terminates
               x8,
       mov
       svc
               0
                      // Call linux to terminate the program
```

.end