

```
//
// Ezekiel Kim
// Lab 8
// Program to Store into variables
// Ezekiel Kim
// /2023
//
.global _start // Provide program starting address

.data
    bA:      .skip    1
    chInit:  .skip    1
    u16Hi:   .hword   0
    wAlt:    .word    0,0,0
    szMsg1:  .skip    10
    dbBig:   .quad    0
    bAv:     .byte    155
    chInv:   .byte    'j'
    u16Hiv:  .hword   88
    wAlt1:   .word    16
    wAlt2:   .word    -1
    wAlt3:   .word    -2
    szMsgv:  .asciz   "And Sally"
    dbBigv:  .quad    9223372036854775807

.text

_start:
// For bA
    ldr     x0,      =bAv      // Value to store in bA
    ldr     x0,      [x0]
    ldr     x1,      =bA       // Address to store bA
    strb    w0,      [x1]      // Store value into bA

// For chInit
    ldr     x0,      =chInv     // Value to store in bA
    ldr     x0,      [x0]
    ldr     x1,      =chInit    // Address to store chInit
    strb    w0,      [x1]      // Store value into chInit

// For u16Hi
    ldr     x0,      =u16Hiv    // Value to store in u16Hi
    ldr     x0,      [x0]
    ldr     x1,      =u16Hi     // Address to store u16Hi
    strh    w0,      [x1]      // Store value into u16Hi

// For wAlt[0]
    ldr     x1,      =wAlt1     // Value to store in wAlt
    ldr     x1,      [x1]
    ldr     x2,      =wAlt2     // Value to store in wAlt
    ldr     x2,      [x2]
    ldr     x3,      =wAlt3     // Value to store in wAlt
    ldr     x3,      [x3]
    ldr     x0,      =wAlt      // Address to store wAlt
    str     x1,      [x0]       // Stores array value
    str     x2,      [x0, #4]   // Stores array value
    str     x3,      [x0, #8]   // Stores array value

// For szMsg
    ldr     x0,      =szMsgv    // Value to store in szMsg
    ldr     x0,      [x0]
    ldr     x1,      =szMsg1    // Address to store szMsg
    str     x0,      [x1]      // Stores value into msg

// For last two bits
    ldr     x0,      =szMsgv    // Value before changing value
    ldr     x1,      =szMsg1    // Value to reference where to store
    add     x0,      x0,      #8
```

```
ldrh    w0,    [x0]
strh    w0,    [x1,    #8]
```

```
// For dbBig
```

```
ldr     x0,    =dbBigv // Value to store in dbBig
ldr     x0,    [x0]
ldr     x1,    =dbBig  // Address to store dbBig
str     x0,    [x1]    // Stored
```

```
// Setup the parameters to exit the program and then call Linux to do it.
```

```
mov     x0,    #0      // Sets return code to 0
mov     x8,    #93     // Service command code 93 terminates
svc     0         // Call linux to terminate the program
.end
```