

```
//
// Ezekiel Kim
// Lab 7
// Program to Load Memory from Data Chunk
// Ezekiel Kim
// /2023
//

.global _start // Provide program starting address

.data
    bA:      .byte    155
    bFlag:   .byte    1
    chInit:  .byte    'j'
    ul6Hi:   .hword   88
    ul6Lo:   .hword   45
    wAlt:    .word     16,-1,-2
    szMsg1:  .asciz    "And Sally and I"
    dbBig:   .quad     9223372036854775807
    szA:     .skip     21
    szFlag:  .skip     21
    szInit:  .skip     21
    szHi:    .skip     21
    szLo:    .skip     21
    szAlt:   .skip     21
    szBig:   .skip     21
    szVar0:  .asciz    "bA"
    szVar1:  .asciz    "bFlag"
    szVar2:  .asciz    "chInit"
    szVar3:  .asciz    "ul6Hi"
    szVar4:  .asciz    "ul6Lo"
    szVar5:  .asciz    "wAlt[0]"
    szVar6:  .asciz    "wAlt[1]"
    szVar7:  .asciz    "wAlt[2]"
    szVar8:  .asciz    "szMsg1"
    szVar9:  .asciz    "dbBig"
    szEq:    .asciz    " = "
    chSq:    .byte     39
    chDq:    .byte     34
    chCr:    .byte     10

.text
_start:
// Load variable label
    ldr     x0,      =szVar0 // load szVar to print
    bl      putstring      // Print it to console
    ldr     x0,      =szEq   // load equal sign
    bl      putstring      // Print it to console
// Print value
    ldr     x1,      =bA     // Load dbA
    ldrb    w0,      [x1]    // Load byte to x0
    ldr     x1,      =szA    // Load string result
    bl      int64asc        // Convert string
    ldr     x0,      =szA    // Load string to x0
    bl      putstring      // Print it to console
    ldr     x0,      =chCr   // Load carriage return
    bl      putchar        // Print carriage return

// Load variable label
    ldr     x0,      =szVar1 // load szVar to print
    bl      putstring      // Print it to console
    ldr     x0,      =szEq   // load equal sign
    bl      putstring      // Print it to console
// Print value
    ldr     x1,      =bFlag  // Load bFlag
    ldrb    w0,      [x1]    // Load byte to x0
    ldr     x1,      =szFlag // Load string result
    bl      int64asc        // Convert string
    ldr     x0,      =szFlag // Load string to x0
```

```
bl    putstring    // Print it to console
ldr   x0,          =chCr    // Load carriage return
bl    putchar      // Print carriage return

// Load variable label
ldr   x0,          =szVar2 // load szVar to print
bl    putstring     // Print it to console
ldr   x0,          =szEq    // load equal sign
bl    putstring     // Print it to console
// Print value
ldr   x0,          =chSq    // Load single quote
bl    putchar      // Print it to console
ldr   x0,          =chInit  // Load chinit
bl    putchar      // Print it to console
ldr   x0,          =chSq    // Load single quote
bl    putchar      // Print it to console
ldr   x0,          =chCr    // Load carriage return
bl    putchar      // Print carriage return

// Load variable label
ldr   x0,          =szVar3 // load szVar to print
bl    putstring     // Print it to console
ldr   x0,          =szEq    // load equal sign
bl    putstring     // Print it to console
// Print value
ldr   x1,          =u16Hi   // Load u16Hi
ldrh  w0,          [x1]     // Load halfword to x0
ldr   x1,          =szHi    // Load string result
bl    int64asc       // Convert string
ldr   x0,          =szHi    // Load string to x0
bl    putstring     // Print it to console
ldr   x0,          =chCr    // Load carriage return
bl    putchar      // Print carriage return

// Load variable label
ldr   x0,          =szVar4 // load szVar to print
bl    putstring     // Print it to console
ldr   x0,          =szEq    // load equal sign
bl    putstring     // Print it to console
// Print value
ldr   x1,          =u16Lo   // Load u16Lo
ldrh  w0,          [x1]     // Load halfword to x0
ldr   x1,          =szLo    // Load string result
bl    int64asc       // Convert string
ldr   x0,          =szLo    // Load string to x0
bl    putstring     // Print it to console
ldr   x0,          =chCr    // Load carriage return
bl    putchar      // Print carriage return

// Load variable label
ldr   x0,          =szVar5 // load szVar to print
bl    putstring     // Print it to console
ldr   x0,          =szEq    // load equal sign
bl    putstring     // Print it to console
// Print value
ldr   x1,          =wAlt    // Load word wAlt[0]
ldrsw x0,          [x1]     // Load word to x0
ldr   x1,          =szAlt   // Load string result
bl    int64asc       // Convert string
ldr   x0,          =szAlt   // Load string to x0
bl    putstring     // Print it to console
ldr   x0,          =chCr    // Load carriage return
bl    putchar      // Print carriage return

// Load variable label
ldr   x0,          =szVar6 // load szVar to print
bl    putstring     // Print it to console
ldr   x0,          =szEq    // load equal sign
```

```
    bl    putstring    // Print it to console
// Print value
    ldr    x1,    =wAlt    // Load word wAlt[1]
    ldrsw  x0,    [x1, #4]    // Load word to x0
    ldr    x1,    =szAlt    // Load string result
    bl    int64asc    // Convert string
    ldr    x0,    =szAlt    // Load string to x0
    bl    putstring    // Print it to console
    ldr    x0,    =chCr    // Load carriage return
    bl    putchar    // Print carriage return

// Load variable label
    ldr    x0,    =szVar7    // load szVar to print
    bl    putstring    // Print it to console
    ldr    x0,    =szEq    // load equal sign
    bl    putstring    // Print it to console
// Print value
    ldr    x1,    =wAlt    // Load word wAlt[2]
    ldrsw  x0,    [x1, #8]    // Load word to x0
    ldr    x1,    =szAlt    // Load string result
    bl    int64asc    // Convert string
    ldr    x0,    =szAlt    // Load string to x0
    bl    putstring    // Print it to console
    ldr    x0,    =chCr    // Load carriage return
    bl    putchar    // Print carriage return

// Load variable label
    ldr    x0,    =szVar8    // load szVar to print
    bl    putstring    // Print it to console
    ldr    x0,    =szEq    // load equal sign
    bl    putstring    // Print it to console
// Print value
    ldr    x0,    =chDq    // Load double quote
    bl    putchar    // Print doublequote
    ldr    x0,    =szMsg1    // Load string to x0
    bl    putstring    // Print it to console
    ldr    x0,    =chDq    // Load double quote
    bl    putchar    // Print double quote
    ldr    x0,    =chCr    // Load carriage return
    bl    putchar    // Print carriage return

// Load variable label
    ldr    x0,    =szVar9    // load szVar to print
    bl    putstring    // Print it to console
    ldr    x0,    =szEq    // load equal sign
    bl    putstring    // Print it to console
// Print value
    ldr    x1,    =dbBig    // Load word wAlt[2]
    ldr    x0,    [x1]    // Load word to x0
    ldr    x1,    =szBig    // Load string result
    bl    int64asc    // Convert string
    ldr    x0,    =szBig    // Load string to x0
    bl    putstring    // Print it to console
    ldr    x0,    =chCr    // Load carriage return
    bl    putchar    // Print carriage return

// 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
```