

```

;-----
;
; Programmer : Larry Aamodt
;
; File name   : hw10_help.s
; Class      : CPTR-215
; Language   : ARM assembly
; Assembler  : Keil
; Target MCU : NXP LPC-2148
; Date Written : 10/27/12
; change history:
;
; Description : Program fragments to demonstrate functionality needed
;              for HW #10
;
; Inputs      :
;
; Outputs     :
;
; NOTES:      In the program for HW10 you will need to initialize
;              a register with the starting address for the source
;              array that contains the string, initialize a register
;              with the address of the destination array, and set a
;              register to zero that will be a counter of the number
;              of characters transferred (note that the byte at the
;              end of the string doesn't need to be counted. It is
;              assumed to be there). Then you will enter a loop that
;              reads a character from the source string, writes it to
;              the destination, and updates pointers and counter.
;
;              NOTE THAT THE HW DEFINITION ASKS YOU TO DESIGN THE
;              PROGRAM WHICH MEANS CREATE A NS DIAGRAM OR FLOW
;              CHART BEFORE WRITING THE PROGRAM CODE.
;-----

```

```

;-----
; CODE
;-----

```

```

        AREA myprogram, CODE, READONLY
        ENTRY
        EXPORT reset_handler

```

```
reset_handler
```

```

; HOW TO GET DATA FROM MEMORY, either ROM or RAM
Main    LDR    r1,=dataX    ;load the address of dataX into register 1
        LDR    r2,[r1]    ;load the contents of dataX into register 2
        ; note that r1 is being used as a pointer
        ; to dataX. dataX is a word and the pointer
        ; is a word, i.e. 32 bits.
        LDR    r1,=varB    ;load the addresses of varB
        LDRB   r2,[r1]    ;load the contents of varB into register 2
        ; note that varB is a byte size variable

        LDR    r1,=array1  ;load address of array1, an array of bytes
        LDRB   r2,[r1]    ;load one byte into r2
        ADD    r1,r1,#1    ;increment r1, the pointer, to access the
        ; next byte in the array

; HOW TO STORE DATA INTO RAM
        LDR    r1,=varY    ;load the address of varY
        STR    r2,[r1]    ;store a word (32 bits) from r2 into memory

        LDR    r1,=varB    ;load the address of varB
        STRB   r2,[r1]    ;store a byte (the right 8 bits) from r2
        ; into memory at the address in r1

stop    B      stop

dataX   DCD    128        ;create one word of data and initialize it
        ; to decimal 128 (could use hex 0x80)
array1  DCB    "Jane",0   ;create an array of 4 bytes with the ascii
        ; characters Jane in it plus a 5th byte
        ; that has a numeric value of zero in it.

;-----
; DATA
;-----
        AREA ram_data, DATA, READWRITE

array2  space  20        ;create an array of 20 bytes into which data
        ; can be written or then read.
varB    DCB    0         ;create space for a byte size variable
varY    DCD    0         ;create space for a word size variable.

;note that you can't place data into the RAM
;variables when you write the program. Only
;the program can place data into RAM when it
;is running.

        END
;-----

```

```
; End of file.  
;  
;  
;
```

Note: If you were to assemble this file you would get a warning on lines 81 and 86 stating:  
Added 3 bytes of padding ....  
This is normal and ok. I'll explain in class.