# CS 219: Homework #10

Due on November 23rd, 2016 at 4:00pm

*Dr. Egbert*

**Matthew J. Berger**

# Problem 1

1.) Write a program called SUB64 to subtract the 64-bit integer in memory locations 0x0150 and 0x154 from the 64-bit integer in 0x0160 and 0x0164. Store the result in memory location 0x0170 and 0x0174.

### Solution:

```
; SUB64
; Template by D. Egbert ver 1.1 11/14/2016
org 100h
section .text
; beginning address of code = 0x0100
;*****************************************************
; put your code here
start:
    mov eax,[0x0150]     ; lower 32 bits of first num
    mov edi,[0x0160]     ; lower 32 bits of second num
    mov edx,[0x0154]     ; lower 32 bits of first num
    mov esi,[0x0164]     ; lower 32 bits of second num
    sub %edi, %eax       ; Subtract lowest 32-bits, borrow reflected in carry flag
    sbb %esi, %edx       ; Subtract highest 32-bits, and the borrow if there was one
    mov [0x170],%esi     ; Copy high bits to memory
    mov [0x174],%edi     ; Copy low bits to memory
; end of your code
;*****************************************************
ILP:    JMP     ILP                                  ;infinite loop
;
TIMES 50H -($-$$) DB 0
section .data
; beginning address of data = 0x0150
;*****************************************************
; put your data items here
; N/A
; end of your data
;*****************************************************
```

# Problem 2

2.) Write a program called COMBINE that combines the low-order nibbles of the four bytes in memory locations 0x0150 to 0x0153 into a single 16-bit word. The nibbles should be ordered low-to-high in the result beginning with the data from location 0x0150. Store the result as 16-bits in memory location 0x0154.

## Solution:

```
; COMBINE
; Templated by D. Egbert ver 1.1 11/14/2016
org 100h
section .text
; beginning address of code = 0x0100
;*****************************************************
; put your code here
start:
                MOV     BL,[0x0150l] ; Store value at 0x0150 in lower bits of reg BX
                MOV     BH,[0x0153h] ; Store value at 0x0154 in higher bits of reg B
                MOV CP,[0x0154]      ; Move offset address to base pointer
                MOV [CP],BX          ; Write value of BX register to memory
; end of your code
;*****************************************************
ILP:    JMP     ILP                                  ;infinite loop
;
TIMES 50H -($-$$) DB 0
section .data
; beginning address of data = 0x0150
;*****************************************************
; put your data items here
; end of your data
;*****************************************************
```

# Problem 3

3.) Write a program called FIND to find the larger of two signed bytes. Assume the two bytes are in memory locations 0x0150 and 0x0151. Store the larger of the two in memory location 0x0152.

### Solution:

```
; Find
; Templated by D. Egbert ver 1.1 11/14/2016
org 100h
section .text
; beginning address of code = 0x0100
;*****************************************************
; put your code here
start:
                MOV     BX,[0x150]
                MOV     CL,0            ; Store larger of two the two bytes
LP1:    MOV     AL,[0x150+1]
                CMP     CL,AL           ; Test if byte 1 > byte 2
                JA      LPC             ; continue testing if not
                MOV     CL,AL           ; otherwise we've found the larger byte
LPC:    DEC     BX
                JGE     LP1
                MOV     [0x152],CL      ; Store the largest byte
; end of your code
;*****************************************************
ILP:    JMP     ILP                             ;infinite loop
;
TIMES 50H -($-$$) DB 0
section .data
; beginning address of data = 0x0150
;*****************************************************
; put your data items here
; end of your data
;*****************************************************
```

# Problem 4

4.) Write a program called LSHIFT to shift logically the 32-bit contents of memory location 0x0150 left according to the 8-bit shift count stored in memory location 0x0154 and store the results at memory address 0x0158.

### Solution:

```
; LSHIFT
; Templated by D. Egbert ver 1.1 11/14/2016
org 100h
section .text
; beginning address of code = 0x0100
;*****************************************************
; put your code here
start:
                MOV BX, [0x0152] ; Store value at 0x0152 in register BX
                MOV CX, 0x0154   ; Move the address for the result into CX
                SHL BX, [0x0158] ; Left shift by the offset stored at 0x0158
                MOV [CX], BX     ; Move the value in BX to memory
; end of your code
;*****************************************************
ILP:    JMP     ILP                                 ;infinite loop
;
TIMES 50H -($-$$) DB 0
section .data
; beginning address of data = 0x0150
;*****************************************************
; put your data items here
; end of your data
;*****************************************************
```

# Problem 5

5.) Write a program called FIND8 to find the largest unsigned 8-bit word in a list. The list begins at address 0x0154. The length of the list is stored in an 8-bit variable at address 0x0150. Store the largest entry in memory location 0x0152.

### Solution:

```
; Find8
; Templated by D. Egbert ver 1.1 11/14/2016
org 100h
section .text
; beginning address of code = 0x0100
;*****************************************************
; put your code here
start:
                MOV     BX,[LSTL]
                MOV     CL,0                    ;USE CL FOR LARGEST VALUE
LP1:    MOV     AL,[BX+DLST]
                CMP     CL,AL                   ;TEST IF BL > AL
                JA      LPC                             ;CONTINUE IF NOT
                MOV     CL,AL                   ;ELSE STORE NEW MAX
LPC:    DEC     BX
                JGE     LP1
                MOV     [MAXV],CL
; end of your code
;*****************************************************
ILP:    JMP     ILP                             ;infinite loop
;
TIMES 50H -($-$$) DB 0
section .data
; beginning address of data = 0x0150
;*****************************************************
; put your data items here
LSTL:   DB      14
MAXV:   DB      0
DLST:   DB      254,5,25,250,100,150,30,200,253,15,23,46,73,175,0
; end of your data
;*****************************************************
```

# Problem 6

6.) Write a program called FIND32 to find the largest unsigned 32-bit word in a list. The list begins at address 0x0160. The length of the list is stored in an 8-bit variable at address 0x0150. Store the largest entry in memory location 0x0154.

## Solution:

```
; Find32
; Templated by D. Egbert ver 1.1 11/14/2016
org 100h
section .text
; beginning address of code = 0x0100
;*****************************************************
; put your code here
start:
                MOV     BX,[LSTL]
                MOV     CL,0                    ;USE CL FOR LARGEST VALUE
LP1:    MOV     AL,[BX+DLST]
                CMP     CL,AL                   ;TEST IF BL > AL
                JA      LPC                             ;CONTINUE IF NOT
                MOV     CL,AL                   ;ELSE STORE NEW MAX
LPC:    DEC     BX
                JGE     LP1
                MOV     [MAXV],CL
; end of your code
;*****************************************************
ILP:    JMP     ILP                             ;infinite loop
;
TIMES 50H -($-$$) DB 0
section .data
; beginning address of data = 0x0150
;*****************************************************
; put your data items here
LSTL:   DB      14
MAXV:   DB      0
DLST:   DB      254,5,25,250,100,150,30,200,253,15,23,46,73,175,0
; end of your data
;*****************************************************
```

# Problem 7

7.) Write a program called SCAN to scan a list of unsigned bytes and find the smallest and largest entries in the list. The length of the list is stored in a 16-bit variable at addresses 0x0152 and 0x0154. The list begins at address 0x0160. Store the smallest byte at address 0x0150 and the largest byte at address 0x0151.

### Solution:

```
; SCAN
; D. Egbert ver 1.1 11/14/2016
org 100h
section .text
; beginning address of code = 0x0100
;*****************************************************
; put your code here
start:
                MOV     BX,[LSTL]
                MOV     CL,0                    ;USE CL FOR LARGEST VALUE
LP1:    MOV     AL,[BX+DLST]
                CMP     CL,AL                   ;TEST IF BL > AL
                JA      LPC                             ;CONTINUE IF NOT
                MOV     CL,AL                   ;ELSE STORE NEW MAX
LPC:    DEC     BX
                JGE     LP1
                MOV     [MAXV],CL
; end of your code
;*****************************************************
ILP:    JMP     ILP                             ;infinite loop
;
TIMES 50H -($-$$) DB 0
section .data
; beginning address of data = 0x0150
;*****************************************************
; put your data items here
LSTL:   DB      14
MAXV:   DB      0
DLST:   DB      254,5,25,250,100,150,30,200,253,15,23,46,73,175,0
; end of your data
;*****************************************************
```

# Problem 8

8.) Write a program called COUNT to count the number of characters in a null-terminated ASCII string that are equal to a KEY. The KEY is stored in memory location 0x0150. The string is stored in memory beginning at address 0x0160. Store the 8-bit count in memory location 0x0154. (Assume the maximum count is 255.)

### Solution:

```
; COUNT
; Templated by D. Egbert ver 1.1 11/14/2016
org 100h
section .text
; beginning address of code = 0x0100
;*****************************************************
; put your code here
start:
                MOV     BX,[LSTL]
                MOV     CL,0                    ;USE CL FOR LARGEST VALUE
LP1:    MOV     AL,[BX+DLST]
                CMP     CL,AL                   ;TEST IF BL > AL
                JA      LPC                             ;CONTINUE IF NOT
                MOV     CL,AL                   ;ELSE STORE NEW MAX
LPC:    DEC     BX
                JGE     LP1
                MOV     [MAXV],CL
; end of your code
;*****************************************************
ILP:    JMP     ILP                             ;infinite loop
;
TIMES 50H -($-$$) DB 0
section .data
; beginning address of data = 0x0150
;*****************************************************
; put your data items here
LSTL:   DB      14
MAXV:   DB      0
DLST:   DB      254,5,25,250,100,150,30,200,253,15,23,46,73,175,0
; end of your data
;*****************************************************
```

# Problem 9

9.) Write a program called ONES to determine the number of bits equal to one in a 32-bit variable. The 32-bit variable is in memory location 0x0154. Store the 8-bit counter in memory location 0x0150.

### Solution:

```
; ONES
; Templated by D. Egbert ver 1.1 11/14/2016
org 100h
section .text
; beginning address of code = 0x0100
;*****************************************************
; put your code here
start:
                MOV     BX,[LSTL]
                MOV     CL,0                    ;USE CL FOR LARGEST VALUE
LP1:    MOV     AL,[BX+DLST]
                CMP     CL,AL                   ;TEST IF BL > AL
                JA      LPC                             ;CONTINUE IF NOT
                MOV     CL,AL                   ;ELSE STORE NEW MAX
LPC:    DEC     BX
                JGE     LP1
                MOV     [MAXV],CL
; end of your code
;*****************************************************
ILP:    JMP     ILP                             ;infinite loop
;
TIMES 50H -($-$$) DB 0
section .data
; beginning address of data = 0x0150
;*****************************************************
; put your data items here
LSTL:   DB      14
MAXV:   DB      0
DLST:   DB      254,5,25,250,100,150,30,200,253,15,23,46,73,175,0
; end of your data
;*****************************************************
```

# Problem 10

10.) Write a subroutine called STRLEN that determines the length of a null-terminated ASCII string. Pass the 16-bit start address of the string to the subroutine in register BX. Return the length, excluding the null byte, in register CX. All registers (except CX) should return to the calling program unchanged.

### Solution:

```
; STRLEN
; Templated by D. Egbert ver 1.1 11/14/2016
org 100h
section .text
; beginning address of code = 0x0100
;*******************************************************
; put your code here
start:
                MOV     BX,[LSTL]
                MOV     CL,0                    ;USE CL FOR LARGEST VALUE
LP1:    MOV     AL,[BX+DLST]
                CMP     CL,AL                   ;TEST IF BL > AL
                JA      LPC                             ;CONTINUE IF NOT
                MOV     CL,AL                   ;ELSE STORE NEW MAX
LPC:    DEC     BX
                JGE     LP1
                MOV     [MAXV],CL
; end of your code
;*******************************************************
ILP:    JMP     ILP                             ;infinite loop
;
TIMES 50H -($-$$) DB 0
section .data
; beginning address of data = 0x0150
;*******************************************************
; put your data items here
LSTL:   DB      14
MAXV:   DB      0
DLST:   DB      254,5,25,250,100,150,30,200,253,15,23,46,73,175,0
; end of your data
;*******************************************************
```

# Problem 11

11.) Write a subroutine called REPLACE that processes a null-terminated string of decimal characters and replaces leading zeros with spaces. Pass the 32-bit address of the string to the subroutine in register BX.

### Solution:

```
; Replace
; Templated by D. Egbert ver 1.1 11/14/2016
org 100h
section .text
; beginning address of code = 0x0100
;*****************************************************
; put your code here
start:
                MOV     BX,[LSTL]
                MOV     CL,0                    ;USE CL FOR LARGEST VALUE
LP1:    MOV     AL,[BX+DLST]
                CMP     CL,AL                   ;TEST IF BL > AL
                JA      LPC                             ;CONTINUE IF NOT
                MOV     CL,AL                   ;ELSE STORE NEW MAX
LPC:    DEC     BX
                JGE     LP1
                MOV     [MAXV],CL
; end of your code
;*****************************************************
ILP:    JMP     ILP                             ;infinite loop
;
TIMES 50H -($-$$) DB 0
section .data
; beginning address of data = 0x0150
;*****************************************************
; put your data items here
LSTL:   DB      14
MAXV:   DB      0
DLST:   DB      254,5,25,250,100,150,30,200,253,15,23,46,73,175,0
; end of your data
;*****************************************************
```

# Problem 12

12.) Write a program called UNPACK to convert the 16-bit BCD variable in memory locations 0x0150 and 0x0151 to four ASCII characters with the high-order digit first, beginning in memory location 0x0154.

## Solution:

```
; UNPACK
; Templated by D. Egbert ver 1.1 11/14/2016
org 100h
section .text
; beginning address of code = 0x0100
;*****************************************************
; put your code here
start:
                MOV     BX,[LSTL]
                MOV     CL,0                    ;USE CL FOR LARGEST VALUE
; end of your code
;*****************************************************
ILP:    JMP     ILP                             ;infinite loop
;
TIMES 50H -($-$$) DB 0
section .data
; beginning address of data = 0x0150
;*****************************************************
; put your data items here
LSTL:   DB      14
MAXV:   DB      0
DLST:   DB      254,5,25,250,100,150,30,200,253,15,23,46,73,175,0
; end of your data
;*****************************************************
```