

2/6/20

# HW4

1. a. .data  
valA BYTE 2  
.code  
mov eax, valA  
sub eax, 3 ; setting sign flag  
add eax, 3 ; resetting sign flag

b. .data  
valA SBYTE -128  
valB SBYTE 127  
.code  
neg valA ; setting overflow flag  
neg valB ; resetting overflow flag

c. .data  
valA BYTE 2  
.code  
mov eax, valA  
sub eax, 2 ; setting zero flag  
add eax, 2 ; resetting zero flag

D.  $4 = +1000$   
 $5 = 0101$   
 $13 = 1101 \leftarrow$  odd # of 1 bits so  
Parity Flag = 0



2. NOT SURE IF "odd elements of the array" REFERS TO INDEX POSITION IN ARRAY OR DECIMAL EQUIVALENT??

ASSUMING INDEX POSITION

```
mov AX, Alpha
add AX, [Alpha+1] ; moving index 1
add AX, [Alpha+3] ; adding index 3
add AX, [Alpha+5] ; adding index 5
mov SUM, AX ; moving total into SUM
```

ASSUMING DECIMAL EQUIVALENT

```
mov AX, Alpha
add AX, [Alpha+1] ; moving 28h = 43
add AX, [Alpha+5] ; moving 6D03 = 27907
mov SUM, AX ; moving total into SUM
```

3. a.  $AX = 4411h$   
b.  $EAX = 76564321h$   
c.  $AX = B4C4h$   
d.  $AX = 0000h$   
e.  $AX = 0000h$

4.  $TYPE\ myarray = 16\ bytes$   
 $length\ of\ myarray = 8$   
 $size\ of\ myarray = 16(8) = 128$   
 $ax = 128$

HW4 (cont.)

2/6/20

- 5.
- a.  $EAX = FFFF344Ch$
  - b.  $EDX = FFFFFFF4Ch$
  - c.  $CX = FFB4h$
  - d.  $EAX = 0000CD23h$
  - e.  $EDX = 00000023h$
  - f.  $CX = 00CDh$

- 6.
- $move\ ax,\ var2$  ;  $ax = 1000h$
  - $move\ ax,\ [var2+4]$  ;  $ax = 3000h$
  - $move\ ax,\ var3$  ;  $ax = -21$  or  $FFEBh$
  - $move\ ax,\ [var3-2]$  ;  $ax = -21$  or  $FFEBh$