

Name: Miandad Khan
ID: 14130
Subject: Micro Processor & Assembly Language
Date: 26/9/2020

Q.1 Use the following variable definitions for the coming question:

```
.data
wVal1      WORD 3000h
wVal2      WORD 7000h
listB      BYTE 50h, 40h, 30h, 20h, 10h
listW      WORD 3000h, 2000h, 1000h
listD      DWORD 30000000h, 20000000h, 10000000h

.code

main PROC
```

(i) What will be the value of the destination operand after each of the following instructions execute in sequence?

```
mov bx, 0ABCDh
movzx eax, bx      ; (a) EAX = 0ABCDh
mov bx, 0DCBAh
movsx eax, bx      ; (b) EAX = 0DCBAh
mov ax, wVal1      ; (c) AX = 3000h
xchg ax, wVal2     ; (d) AX = 7000h
                  ;(e) val2 =3000h
```

```
mov wVal1, ax      ; (f) val1 = 7000h
mov al, listB      ; (g) AL = 10h
mov al, [listB+4]  ; (h) AL = 50h
mov ax, listW      ; (i) AX = 5000h
mov ax, [listW+4]  ; (j) AX =
mov eax, listD     ; (k) EAX =
mov eax, [listD+8] ; (l) EAX =
```

Q2 Write down the values of the Carry, Sign, Zero, and Overflow flags after each instruction has executed:

```
mov ax, 7FF0h add al, 10h      ; (a) CF = 1 SF = 0 ZF = 1 OF = 0
add ah, 1                      ; (b) CF = 0 SF = 1 ZF = 0 OF = 1
add ax, 2                      ; (c) CF = 0 SF = 1 ZF = 0 OF = 0
mov al, 1 sub al, 2            ; (d) CF = 1 SF = 0 ZF = 1 OF = 0
```

Q.3 Use the following data definitions for the coming question:

```
.data
listB      BYTE 60h, 50h, 40h, 30h, 20h, 10h
listW     WORD 4 DUP(?), 1000h
string1   BYTE "Assembly Language", 0
```

(i) What will be the value of EAX after each of the following instructions execute?

```
mov eax, TYPE listB          ; (a) EAX = 1
mov eax, LENGTHOF listB     ; (b) EAX = 6
```

mov eax, SIZEOF listB ; (c) EAX = 6
mov eax, TYPE listW ; (d) EAX = 2
mov eax, LENGTHOF listW ; (e) EAX = 4
mov eax, SIZEOF listW ; (f) EAX = 8
mov eax, SIZEOF string1 ; (g) EAX = 17

(i) Write an instruction that moves all four bytes in listB to the EAX register.

Ans:-

```
move ax,DWORD PTR listB
```

(ii) Insert a LABEL directive in the given data that permits listB to be moved directly to EAX register.

Ans:-

```
listB LABEL DWARD  
listB WORD 3 DUP(?),2000h  
  
.data  
  
move ax,listB
```

Q.4 Use the following data definitions for coming question:

listB BYTE 10h, 20h, 30h, 40h
listW WORD 8Ah, 3Bh, 72h, 44h, 66h
listD DWORD 1, 2, 3, 4, 5
pointer1 DWORD listD

What will be the value of the destination operand after each of the following instructions execute in sequence?

```

mov esi, OFFSET listB
mov al, [esi] ; (a) AL = 10h
mov al, [esi+3] ; (b) AL = 40h
mov esi, OFFSET listW + 2
mov ax, [esi] ; (c) AX = 003Bh

mov edi, 8
mov edx, [listD + edi] ; (d) EDX = 3
mov edx, listD [edi] ; (e) EDX = 3
mov ebx, pointer1
mov eax, [ebx+4] ; (f) EAX = 2

```

Q.5 Implement the following pseudocode in assembly language:

```

(i)   if( var1 <= var2 )
      var3 = 15;
      else {
      var3 = 10; var4 = 30;
      }

```

Ans:-

```

mov eax,var 1
cmp eax,var2
Jle L1
mov var 3,10
mov var 4,30
jmp L2
L1: mov var3,15

```

L2:

- (ii) **if (val1 > ecx) AND (ecx > edx) then**
 A = 12
 else
 B = 6;

Ans:-

```
cmp val1,ecx
jna else
cmp ecx,edx
jna else
mov A,12
```

```
jmp next
else: mov B,6
next:
```

- (iii) **while(ebx < eax)**
 ebx = ebx + 1;

Ans:-
cmp ebx,ebx+1
jae nxt
inc ebx
jmp top
next:

Q.6 (i) What will be the final value of EAX in this example?

```
        mov eax, 0
        mov ecx, 10
L1:     mov eax, 3
        mov ecx, 5
L2:     add eax, 5
        loop L2
        loop L1
```

Ans:-

The answer is “Infinite” of this loop.

(ii) Write a program that calculates the following expression, using registers:

$$A = (A + B) - (C + D)$$

Ans:-

```
.model flat,stdcall
.stack 4096
ExitProcess PROTO, dwExitCode:DWORD
.code
Main PROC
mov eax,3h
mov ebx,8h
mov ecx,1h
move dx,8h
add eax,ebx
sub eax,ecx
```

```
INVOKE EXITPROCESS,0
```

```
main ENDP
```

```
END main
```

(iii) Write a program that uses a loop to copy all the elements from an unsigned Word array into an unsigned doubleword array.

Ans:-

```
.model flat,stdcall
```

```
.stack 4096
```

```
ExitProcess PROTO, dwExitCode:DWORD
```

```
newArray DWORD LENGTHOF array DUP(?)
```

```
.code
```

```
main PROC
```

```
move cx,LENGTHOF array
```

```
mov ESI,OFFSET array
```

```
mov EDI,OFFSET newArray
```

L1:

```
MOV EAX,0
```

```
MOV AX,[ESI]
```

```
MOV[EDI],EAX
```

```
ADD ESI,TYPE array
```

```
ADD EDI, TYPE newArray
```

L1

```
INVOKE ExitProcess,0
```

```
main ENDP
```

```
END main
```

- (iv) Write a program that displays a string in all possible combinations of foreground and background colors (16 x 16 =256). The colors are numbered from 0 to 15, so you can use a nested loop to generate all possible combinations. Also use a delay of 1s in each foreground color change.

Ans:-

```
INCLUDE Irvine32.inc
```

```
.data
```

```
count DWORD ?
```

```
.code
```

```
Main PROC
```

```
mov eax,0+(0*16)
```

```
mov cx,16
```

```
L1:
```

```
mov count,ecx
```

```
push eax
```

```
mov cx,16
```

```
L2:
```

```
call SetTextColor
```

```
push eax
```

```
mov al,'H'
```

```
call WriteChar
```

```
pop eax
```

```
inc eax
```

```
LOOP L2
```

```
Call Crif
```



```
Pop eax
Add eax,16
mov ecx ,count
LOOP L:
call crif
call WaitingMsg
exit
main ENDP
```