(1)

NAME ~ UBAID ALI

ID ~

Semester ~ 8th

Department ~

Assignment ~

Answers:~

1) and ax, 00FFh

2) or ax, 0FF00h

3) xor eax, 0FFFFFFFFh

4) test eax, 1 ;(low bit set if eax is odd)

5) or al 00100000b

26) and al, 00001111b

22) data

memval DWORD?

- code

mov al, BYTE PTR memval
xor al, BYTE PTR memval+1
xor al, BYTE PTR memval+2
xor al, BYTE PTR memval+3

6) JA, JNBE, JAE, JNB,
JB JNAE, JBE, JNA

7) JG, JNLE, JGE, JNL, JL,
JNGE, JLE, JNG

8) No because the JG is used with signed value and (8 bit is negative, and 264 is positive)

24) Yes

25) Yes (the unsigned representation of -42 is compound to 26)

Push esi
Push ecx

3½

28) cmp dx, cx
Jbe L1

29) cmp ax, cx
Jg L2

30) and al, 111 111 00b
JZ L3
JMP L4

16) (a) 011 011 01
(b) 0100 1000
(c) 011 011 111
(d) 101 000 11

18) (a) CF=0, ZF=0, SF=0
(b) CF=0, ZF=0 SF=0
(c) CF=1, ZF=0, SF=1

19) JECX 2

(4)

17) (a) 85h

(b) 84h

(c) BFh

(d) FEh

34) INCLUDE Irvine32 INC

N = 10

.data

array SDWORD N DUP(-10, -8, -6, -4

-2, -1, 1, 3, 52)

J DWORD ?

K DWORD ?

.code

main PROC

call clrscr

~~call~~ mov J, 0

(5)

```
mov K, 10
mov ESI, OFFSET array
mov ECX, N
call Summing Array Element in Range
call with Int
call crlf
mov J, -10
mov K, 0
mov ESI, OFFSET array
mov ECX, N
call Summing Array Elements in Range
call WriteInt
call crlf
call WaitMsg
exit
```

main ENDP

Summing Array Elements in Range Proc

Push ecx.

Push esi.

mov eax, 0

l1:

mov ebx, [esi]

comp ebx, j

jle true1

Jmp next

true1:

comp ebx, k

Jle true2

Jmp next

true2

add eax ebx

next:

add eax, [ebx esi, 4

Loop l

Pop esi

Pop ecx

ret

summing Array Elementes in Range ENDP

END main

3) include Irvine 32 line

.data

byte 1 BYTE 11111110b, 110 1110b,

1000 111b, 11001100b, 11001010b,

11001010b, 11001010b, 1100

byte 2 BYTE 11111110b, 11011111b,

10001110b, 11010010b, 11001100b,

(8)

11001011b, 11001010b, 11001010b.
1100

```
.code
    main PROC
    mov esi, OFFSET
    mov ecx, SIZEOF byte 2
    call PFcheck
    call writeint
    mov esi, OFFSET byte 2
    mov ecx, SIZEOF 2
    call PF.check
    call write int
    exit
    main ENDP
PFcheck PROC
; eax PF=1 TRUE PF=0 False
; esi ecx
```

```
Push    esi
Push    ecx

Sub ecx, 1
mov al, c
xor al, 0

mov al, (esi)

L1 :

inc esi
xor al, (esi)

mov bl, (esi)
Loop L1

JP LPE 1
mov eax, 0
JMP LEND
LPE 1:

mov eax, 1
LEND
PoP ecx
PoP esi
ret
        PF check ENDP
        END main
```

2] INCLUDE32.INC

```
.data
        N DWORD 10
        A DWORD 7
        B DWORD 8

. code
        main PROC

        mov eax, N
        mov ebx, A
        mov ecx, B

TOP:

        cmp eax, 0
        Jbe Next
        cmp eax, 3
        Jne L1
        jmp L4

L1:

        cmp eax, ebx

        jb L3
        ja L2
```

```
L2:
        cmp  eax, ecx
        ja   L3
        jb   L4

L3
        sub  eax, 2
        JMP  TOP

L4
        Sub  eax, 1
        jmp  top

Next
        invoke Exit Process, 0

main    endp
end     main
```

3)(a)

```
cmp  EBX , ECX
JBE  L1, if (ebx<c = ecx)
cmp  EBX , Val 1
JBE  L1
MOV  X, 1
JMP  L2
```

L1 mov x, else, x=2

L2

(b) CMP EBX, ECX

JBE L1

CMP EBX, EDX

JBE L1

JMP L3, both true, go to L2

L1 CMP EDX, EAX

JBE L3; if (edx<=eax), go to L3

L2 MOV X, 1

c) BX = 006Bh

BX = 092h

BX = 064BBh

BX = A857h

EBX = BFAFF69Fh