:: ASSIGNMENT # 4 ::

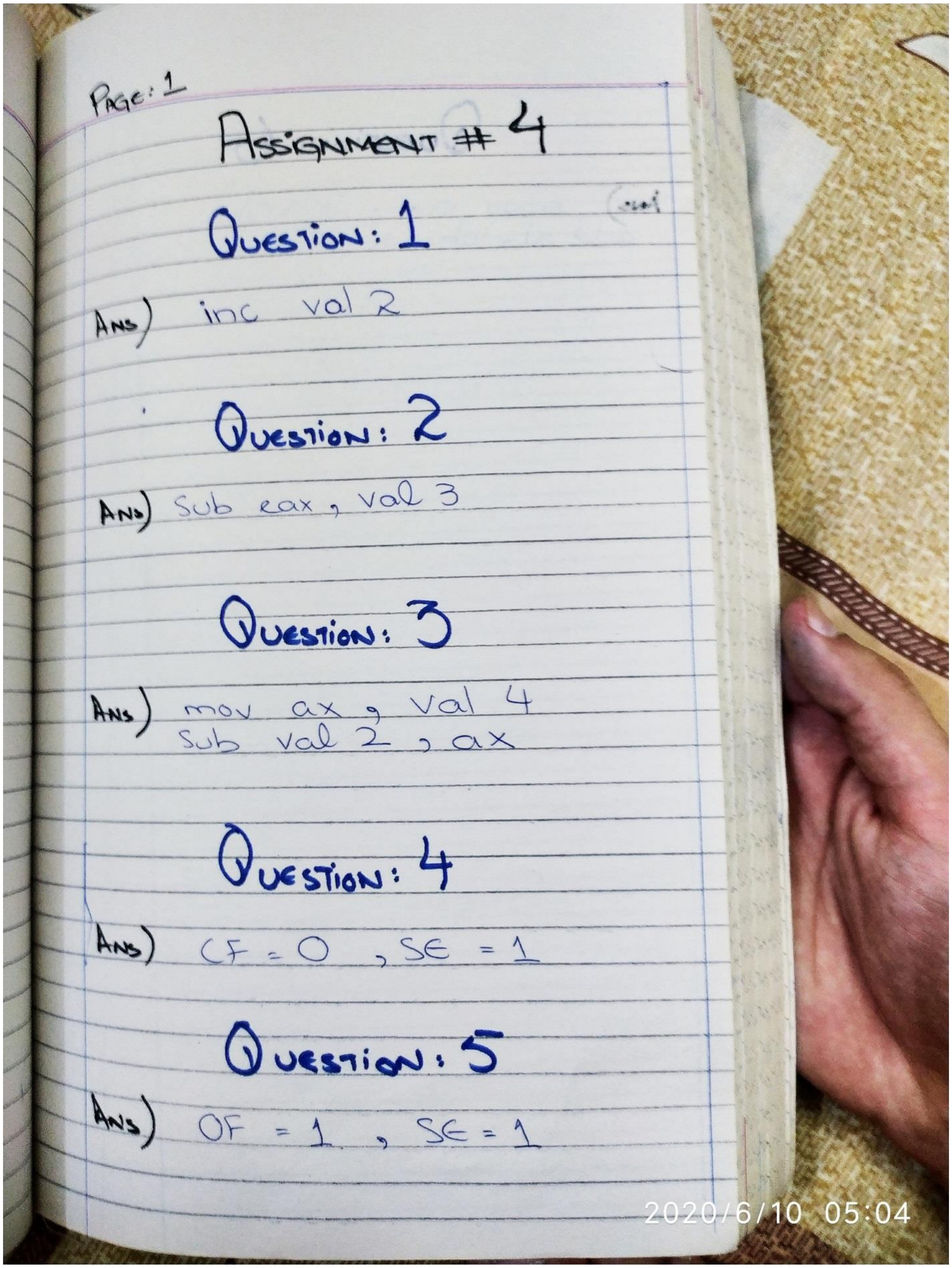
ID: 11533

Name: Ashir Ali Khan

Subject: Microprocessor and Assembly Language

Teacher: Sir Muhammad Amin

Iqra National University



Page: 1

Assignment # 4

Question: 1

Ans) inc val2

Question: 2

Ans) Sub eax, val3

Question: 3

Ans)
```
mov ax, val4
Sub val2, ax
```

Question: 4

Ans) CF = 0 , SF = 1

Question: 5

Ans) OF = 1 , SF = 1

2020/6/10 05:04

Question: 6

Ans) mov ax, 7FF0h
add al, 10h; a: CF = 1 SE = 0 ZF = 1 OF = 0
add oh, 1; b: CF = 0 SE = 1 ZF = 0 OF = 1
add ax, 2; c: CF = 0 SE = 0 ZF = 0 OF = 0

Question: 7

Ans) mov esi, OFFSET my Bytes
mov al, [esi] ; a. AL = 10h
mov al, [esi+3] ; b. AL = 40h
mov esi, OFFSET my word +2 ; C. Ax = 003Bh
mov edi, 8

mov edx, [my Doubles +edi] ; d. EDX = 3
mov edx, my Doubles [edi] ; e. EDX = 3
mov ebx, my Pointer

mov eax, [ebx + 4] ; F. EAX = 2

Question: 8

Ans) mov esi, OFFSET my Bytes

mov ax, [esi] ; a. Ax = 2010h
mov eax, DWORD PTR myWords ; b. EAX =
x mov ax, [esi] x
[illegible]
x mov esi, my WORD x

2020/6/10 05:04

```
mov esi, myPointer   ;
mov ax, [esi+2]      ; c. Ax = 0000
mov ax, [esi+6]      ; d. Ax = 0000
mov ax, [esi-4]      ; e. Ax = 0044h
```

Question: 9

Ans) The program doesnot stop because;
First loop instruction decreement ECX to zero. The second loop instruction decrements ECX to FFFFFFFFh, causing the outer loop to repeat.

Question: 10

Ans)
```
.DATA
      Count DWORD = ?
   .CODE
  mov eax, 0
  mov ecx, 10 ; outer loop Counter
 L1 :
mov count, ecx
mov eax, 3
mov ecx, 5 ; inner loop counter
L2 :
 add eax, 5
loop L2 ; repeat inner loop
mov ecx, count
loop L1 ; repeat outer loop
```

Question: 11

Ans)
```
mov ax, word ptr three
mov bx, word ptr three+2
mov three , bx
mov word ptr three+2, ax
```

Question: 12

Ans)
```
xchg A , B
xchg A , C
xchg A , D
```

Question: 13

Ans) * Parity flag (PF) will be set if there is an even number of 1 bits in the message byte.

* parity flag (PF) will be zero for the message byte having an odd number of 1 bits.

• Code
```
mov al, 01110101 b
add al, 0000000 ; AL-0111 0101,
                        PF-0
```

After the execution of the ADD instruction, AL contain the value of the message byte since there are five (5) odd number of ones in the AL register -
Thus, PF = 0

Question: 14

Ans) Any non-zero operand causes the carry flag to be set.

Example:-

```
• DATA
val B  BYTE 1, 0
val C  SBYTE -128

• CODE
neg val B      ; CF = 1 , OF = 0
neg [val B+1]  ; CF = 0 , OF = 0
neg val C      ; CF = 1 , OF = 1
```

Question: 15

Ans)
```
mov al , 0FFh
add al, 1
```

Question: 16

Ans)
```
mov al, 0FFh
add al, 1 ; CF = 1, AL = 00
; Try to go below zero:
mov al, 0
sub al, 1 ; CF = 1, AL = FF
```

Question: 17

Ans) Solution:—

```
INCLUDE Irvine32.inc

.DATA
val1 SDWORD 8
val2 SDWORD -15
val3 SDWORD 20

Find val SDWORD = ?

.Code
main PROC

mov eax val2
neg eax ; eax = -15
add eax, 7 ; -val2 + 7

mov ebx, val3
add ebx, val1 ; val3 + val1
```

```
mov
call Dum
exit
main ENDP
END main
```

Questi

Ans)
```
.DATA
intarray DW
30000h,

.Code

main proc

mov edi, OFFS
mov ecx, LENG
mov eax, 0

L1

add eax
add edi
Loop L1
invo
```

2020/6/10 05:05

```
    sub eax , ebx

mov final val , eax
call DumpRegs ; Display the registers

exit
main ENDP
END main
```

Question: 18

Ans) • DATA

```
intarray DWORD 10000h , 20000h,

   30000h , 40000h
```

• Code

```
  main proc

mov edi , OFFSET ~~intarray~~
mov ecx, LENGTHOF intarray
mov eax, 0

L1
  add eax , [edi]
  add edi , TYPE intarray
  Loop L1

invoke ExitProcess , 0

  main endp
  end main
```

2020/6/10 05:05

Question : 19

Ans)
```
mov al , 80h
add al , 80h
```

Question : 20

Ans)
```
mov al, OFFh
inc al
jZ INC - overflow

mov al , 1
dec bl
jZ DEC - overflow

INC - overflow
DEC - Overflow
```

Question : 21

Ans)
```
mov eax, TYPE myBytes ; a.1
mov eax, LENGTH OF myBytes ; b. 4
mov eax, SIZEOF myBytes ; c. 4
mov eax, TYPE myword ; d. 2
mov eax, LENGTH OF myWord ; e. 4
mov eax, SIZEOF my Word ; f. 8
mov eax, SIZEOF my String ; g. 5
```

Question: 22

Ans) mov dx, WORD PTR myBytes

Question: 23

Ans) mov al, BYTE PTR myWords +1

Question: 24

Ans) mov eax, DWORD PTR myBytes

Question: 25

Ans) myWords LABEL DWORD
myWords WORD 3 DUP(?), 2000h

• data
mov eax, ~~my~~words D

Question: 26

Ans) •Data

myByte BYTE 10h, 20h, 30h, 40h

myWords WORD 3 DUP(?). 200h

2020/6/10 05:05

Page. 10

```
myWords DLABEL DWORD
myWords WORD 3 DUP(?), [illegible]
                          2000h

• Code
mov eax, myWords D
```

QUESTION: 27

ANS) Programming Name: bigEndian to Little Endian

```
• 386
model Flat, stdcall
• Stack 4096
ExitProcess PROTO, dwExitcode : DWORD

• data
bigEndian BYTE 12h, 34h, 56h, 78h
little Endian DWORD ?

• Code
 main PROC
mov al, [bigEndian+3]
mov BYTE PTR [little Endian], al

mov al, [big Endian +2]
mov BYTE PTR [little Endian+1], al

mov al, [bigEndian+1]

mov BYTE, PTR [little Endian+2], al
```

```
INVOKE Ex
main ENDP
END main
```

QUESTION: 2

```
ANS) • 386
·model Flat, S
·Stack 4096
ExitProcess PROTO

•data
array WORD 0,
newArray DWORD LENG

•Code
main PROC

mov ecx, LENGTHOF
mov ESI, OFFSET
mov EDI, OFFSET
L1:
Mov EAX
Mov A
```

2020/6/10 05:05

```
mov al, [big Endian]
mov BYTE PTR [little Endian +3], al

INVOKE ExitProcess, 0

main ENDP

END main
```

QUESTION: 28

Ans)
```
.386
.model flat, std call
.stack 4096
ExitProcess PROTO, dwExitCode:DWORD

.data
    array WORD 0,2,5,9,10
newArray DWORD LENGTHOF array DUP(?)

.code
  main PROC

mov ecx, 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
```

2020/6/10 05:05

```
Loop L1

INVOKE ExitProcess, 0
main ENDP
END main
```

QUESTION : 29

Ans) Solution:-

```
.386
.model flat, stdcall
.stack 4096
ExitProcess PROTO, dwExitcode :DWORD

.data
decimalArray DWORD 1,2,3,4,5,6,7,8

.code

main PROC

Mov ESI, OFFSET decimal Array
Mov EDI, OFFSET decimal Array
mov ecx, LENGTHOF decimal Array - 1

L1:
    ADD EDI, TYPE decimal Array
Loop L1

mov ecx ; LENGTHOF decimal Array
```

```
L2:
  Mov EAX , [ESI]
  Mov EBX , [EDI]
  XCHG EAX , EBX
  Mov [ESI] , EAX
  Mov [EDI] , EBx

 ADD ESI , TYPE decimal Array
 Sub EDI , TYPE decimal Array
 DEC ECX

 Loop L2

  INVOKE Exit Process , 0
  main ENDP
  END main
```

Question: 30

Ans) Solution :-

```
 .386
 .model flat , stdcall
 .Stack 4096
 ExitProcess PROTO, dwExitCode :DWORD

.data
Source BYTE "This is the Source String",0
target BYTE SIZE Of Source DUP ('#')

.code
```

2020/6/10 05:05

Page: 14

```
  main PROC
  mov esi, 0
  mov edi, LENGTHOF source - 1
  mov ecx, SIZE OF source

L1:
  mov eax, 0
  mov edial, source [esi]
  mov target [edi], al
  inc esi
   dec edi
   loop L1

INVOKE ExitProcess, 0
main ENDP
 END main
```

ANSWERS

Ans) and

Ans) Or

Ans) XOr

Ans) test

Ans) Or

Ans) JA

JR