nal University of Computer and Emerging Sciences 0110

FAST School of Computing Fall-2023 Islamabad Campus

EE-2003 Computer Organization and Assembly Language

Serial No:

Sessional Exam-II

Total Time: 1 Hour

Total Marks: 60

Tuesday, 7th November 2023.

Signature of Invigilator

Course Instructor

Mr. Aqib Rehman, Mr. Taimur Shahzad, Mr. Obaid Ullah, Mr. Shams Farooq, Mr. Farrukh Bashir

Areeqa Malik 22i-1167 B

Student Name Roll No. Section Signature

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

Instructions:

1. Attempt all questions on the question-book. Read the question carefully, understand the question, and then attempt it.
2. No additional sheet will be provided for rough work. Use the provided space for rough work
3. After asked to commence the exam, please verify that you have **Eight(8)** different printed pages including this title page. There are a total of **5** questions.
4. Calculator sharing is strictly prohibited.
5. Use permanent ink pens only. Any part done using soft pencil will not be marked and cannot be claimed for rechecking.

	Q-1	Q-2	Q-3	Q-4	Q-5	Total
Marks Obtained	10	05	10	10	18	53
Total Marks	10	10	10	10	20	60

Good.

In Q#2 : 111111 this result is circled, but it seems right to me, After A=A-M, so can you please recheck this Question.

Regards!

(immediate steps wrong)

National University of Computer and Emerging Sciences

FAST School of Computing   Fall-2023   Islamabad Campus

Question 1 [3+7= 10 Marks]

i. Make corrections to the bubble sort code written for 16-bit processor. [3]

ii. You are supposed to modify the given code for 32-bit processor for declaration given on the left. [7]

a. Your array should be in descending order.

b. Array should order work for sign numbers as well.

16-Bit Processor

```
.data
     array db -5, 1, 0, 3
     swap db 0
.code
 mov cx, 4
 dec cx
start:
      mov swap, 0
      mov bx, 0

loop1:
          al
      mov ax, [bx+array]
      cmp ax, al [bx+array+1]
      jbe noswap
          dl
      mov dx, [bx+array+1]
      mov [bx+array+1], ax al
      mov [bx+array], dx dl
      mov swap, 1

noswap:

      add bx, 2 1
      cmp bx, cx
      jne loop1

      cmp swap, 1
      je start
main ENDP
END main
```

3

Correction! 0

32-Bit Processor

```
.data
  array dword -5, 1, 0, 3, 4, -2
  swap db 0
.code
  mov ecx, 6
   dec ecx
start:
      mov swap, 0
      mov ebx, 0
loop1:
     mov eax, [ebx+array]
     cmp eax, [ebx+array+4]
     jbe noswap
     mov edx, [ebx+array+4]
     mov [ebx+array+4], eax
     mov [ebx+ array], edx
     mov swap, 1
noswap:
     add ebx, 4
     cmp ebx, ecx
     jne loop1
     cmp swap, 1
     je start
main ENDP
END main
```

7

Signed Jump Flags

CMP Results	Flags
Destination < source	SF ≠ OF
Destination > source	SF = OF
Destination = source	ZF = 1

Unsigned Jump Flags

CMP Results	ZF	CF
Destination < source	0	1
Destination > source	0	0
Destination = source	1	0

National University of Computer and Emerging Sciences 0110

FAST School of Computing Fall-2023 Islamabad Campus

Question 2 [8+2=10 Marks]

You are supposed to divide $(10100)_2/(11)_2$ using unsigned division. Check your answer by performing binary division.

05

Division

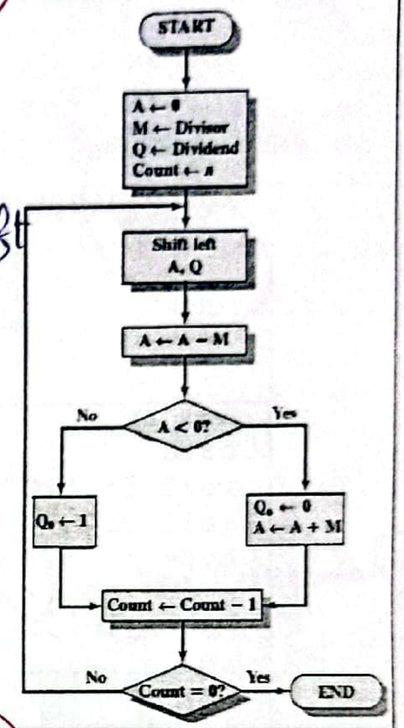
M 000011 -M 111101

A	Q	Steps	Count
000000	010100	Initiaize	6

	A	Q	
	000000	101000	Shift left
A<0	111101	101000	Sub
	000000	101000	$Q_0=0$ $A \leftarrow A+M$
	000001	010000	Shift left
A<0	111110	010000	Sub
	000001	010000	$Q_0=0$ $A \leftarrow A+M$
	000010	100000	Shift left
A<0	111111	100000	Sub
	000010	100000	$Q_0=0$ $A \leftarrow A+M$
	000001	000000	left
A>0	000010	000000	Sub
	000010	000001	$Q_0=1$
	000100	000010	left
	000001	000010	Sub
A>0	000001	000011	$Q_0=1$
	000010	000110	left
A<0	111111	000110	Sub
	000010	000110	$Q_0=0$ $A \leftarrow A+M$

Quotient	000110 (6)	Remainder	000010 (2)

FLOW CHART



Answer Check By Binary Division

```
      110
11 | 10100
     11
     ----
      11
      11
     ----
      10
```

Question 3 [2+2+6=10 Marks]

Consider the given assembly code. For part I,II & III fill register & for part IV fill memory(marks will be given for filling value of s3.

Fill Memory and Registers

```
;Part I
.code
 mov ax,0
 mov al,'9'
 add al,'8'
 aaa
 or ax,3030h
```

2

ax	0
al	39
al	71
ax	0107
ax	3137

```
;Part II
.data
 val1 db '5'
 val2 db '7'
.code
 mov ax,0
 mov al,val1
 add al,val2
 aas
 or al,30h
```

1 -

ax	0000
al	35
al	6C
ax	FF09
al	FF39

```
;Part III

.data
 val1 db 5h
 val2 db 6h
.code

 mov bl,val1
 mov al,val2
 mul bl
 aam
```

1

bl	05
al	06
bl	05
ax	0300

```
;Part IV
.data
   string1 BYTE "987"
   string2 BYTE "789"
   s3 dB (SIZEOF string1+1)DUP(0),0

.code
main Proc

mov esi,SIZEOF string1 - 1
mov edi,SIZEOF string1
mov ecx,SIZEOF string1
mov bh,0

L1:
  mov ah,0
  mov al, string1[esi]
  add al,bh
  aaa
  mov bh,ah
  add al,string2[esi]
  aaa
  or bh,ah
  or bh,30h
  or al,30h
  mov s3[edi],al
  dec esi
  dec edi
loop L1

mov s3[edi],bh

main ENDP
END main
```

ah=0,
al='7',
al='7
ah=0, 07
al=

bh=0, ah
al=07
bh=0
bh=30
al=37

edi/ecx

9 8 7 esi↓ 7 8 8 [s3 0 0 . 0

6

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0100	39	38	37	37	38	39	31	37	37	00						
0130																

Question 4 [10 Marks]

Write down the value of the requested register/variable after the execution of each specified line.

10

Code

```
.data
    mFLAGS WORD 23997 -> 0101110110111101
    mDATA BYTE 10101010b
    permTable DB "31302928"

.code
main PROC
    movzx eax, mFLAGS
    test mFLAGS, 0000000000000001b
    JZ skip

    mov esi, offset permTable
    mov ecx, 32
    mov ebx, 1
    mov edx,0

    mov dh, byte ptr [esi]     ) dh= 3
    inc esi
    mov dl, byte ptr [esi]     ) dl= 1
    SUB dh, 48                 dh=3
    SUB dl, 48                 dl=1
dl=3 mov al, dh
    mov dh, 10   -> dh,10
    mul dh
    add dl, al

    MOV al, dl   (dl=31)
    mov ecx, 2
    mov al, mDATA             ; Load mData into al
    mov dl, 8                 ; Number of bits in mData

SwapLoop:
    shl al, 1 ; Shift left (MSB goes to carry flag)
    rcr mDATA, 1 ; Rotate right through carry in mData
    loop SwapLoop ;
    mov ecx, 1

    rcr mDATA, cl
    01101010, 1
CF=1
    rol mDATA, 1   10110101, E
skip:

    INVOKE ExitProcess, 0

main ENDP
END main
```

Rogh Work

1010101 0
CF
01010100
RCR
~~10101010~~
Mdata = 10101010
CF = 1
11010101
mdata ← 01101010
CF=1

Value

Jump Taken/Not

~~NOT~~

DL (decimal)

31 or 1E

;Demonstrate loop rough work

mDATA

01101010

mDATA

10110101

mDATA

11011010

Question 5 [20 Marks]

i. SHL instruction performs unsigned multiplication when the multiplier is a power of 2 and any other number can be expressed in powers of 2. Write instructions to find the product of **AL** by **29**. Not supposed to use any type of mul instruction or algorithm write an optimized code

As $29 = \underbrace{2^4} + \underbrace{2^3} + \underbrace{2^2} + 2^0$

```
mov al, 29
mov bl, al
mov cl, al
mov dl, al
shl bl, 4  ; (2^4)
shl cl, 3  ; (2^3)
shl al, 2  ; (2^2)
add bl, cl
add bl, dl  →  inc bl
                mov al, bl ; (al = al * 29 *
```

05

ii. Write a generic program that count number of 1's in a declaration given below. Ary can be of any size or type

```
.Data
    ary dq 0ABCDEF1234567890H,-1,1929394959697900h,-2
    NumberOfOnes dw 0
```

```
.Code
 mov ecx, lengthof ary
 mov edx, 0
 mov esi, offset NumberOfOnes
Loop1:
  cmp [ary+edx], 1
  jnz skip
   add [esi], 1
 skip:
  add edx, type ary
 Loop loop1;
```

esi

02

iii. Update Flags after executing following code and state which of the following jumps will be taken or not taken? Mention flags required for the jump. No marks for direct answer? Rough work is required.

CODE

```
mov al,24        (00011000)
mov bl,-25       (11100111)
cmp al,bl

ja l1
l1: jnb l2
l2: jg  l3
l3: jle l4
l4: jnae  l5
l5:
```

04

ROUGH SPACE 1010 10 1010

CF=1 000 11 00 0
− 111 00111
(00110001)

	Taken	Not Taken	Flags Required
ja		✓	Carry, Zero
jnb		✓	carry, Zero
jg	✓		Sign, Overflow
jle		✓	Zero, sign, overflow
jnae	✓		Zero, carry

Flags				
Flags	Sign	0	Overflow	0
	Zero	0	Parity	0
	Carry	1	Auxiliary	1

iv. Update Register after every instruction

```
.code

mov ax,0DA15h
mov bl,025H
xor al,-1
xor bl,al
xor al,bl
xor ah,ah
clc
mov al,05AH
ROR al,4
RCL al,1
SAL al,2
SAR al,2
```

1010 0001 0101
1101
0010 0101
0101 1010
1010 0101 CF=1
0100 1011

05

al	1110 1010 (EA)
bl	1100 1111 (CF)
al	0010 0101 (25)
ah	000 0000 (00)
al	1010 0101 (A5)
al	0100 1011 (4B)
al	0010 1100 (2C)
al	0000 1011 (0B)

As you being the youth. How could you contribute to make Pakistan a better place? [mufta]

As youth of Pakistan, we can contribute to country by studying well, and after studies put effort for property of country.

02