National University of Computer and Emerging Sciences

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

_____ _____

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						
Total Marks	**10**	**10**	**10**	**10**	**20**	**60**

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:

      mov al, [bx+array]
      cmp al, [bx+array+1]
      jbe noswap

      mov dl, [bx+array+1]
      mov [bx+array+1], al
      mov [bx+array], dl
      mov swap, 1

noswap:

      add bx, 1
      cmp bx, cx
      jne loop1

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

32-Bit Processor

```
.data
  array dword -5, 1, 0, 3, 4, -2
  swap db 0
.code
main Proc
 mov ecx, 24 or  sizeof array
 sub ecx, 4

 start:
      mov swap, 0
      mov ebx, 0
loop1:
      mov eax, [ebx+array]
      cmp eax, [ebx+array+4]
      jle 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
```

Signed Jump Flags

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

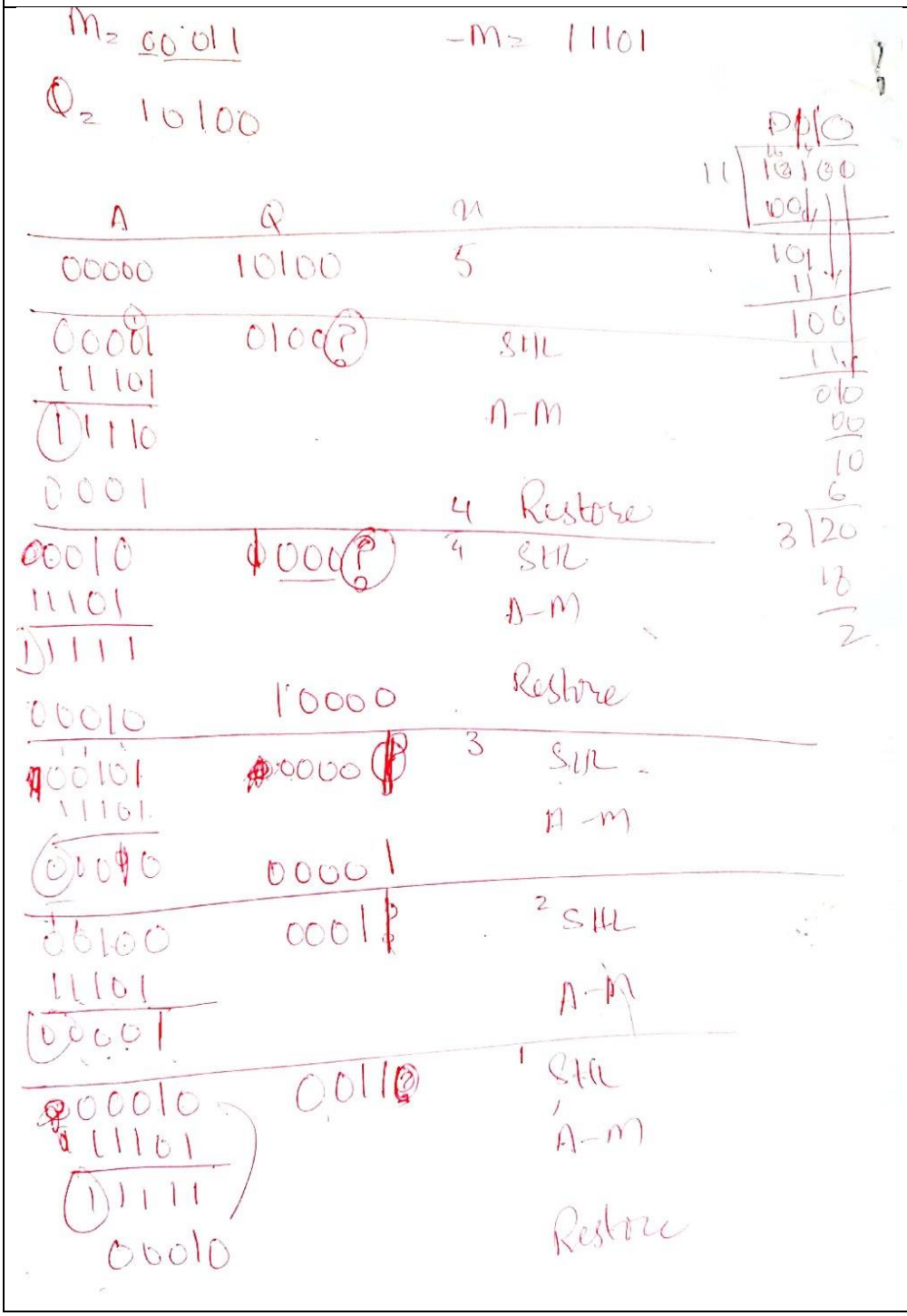
Unsigned Jump Flags

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

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.

Division



Quotient	00110	Remainder	00010

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
```

ax	0000
al	39
al	71
ax	0107
ax	3137

;Part II

```
.data
 val1 db '5'
 val2 db '7'
.code
 mov ax,0
 mov al,val1
 sub al,val2
 aas
 or al,30h
```

ax	0000
al	35
al	FE
ax	FF08
al	38

ERROR Suppose to be sub

;Part III

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

 mov ax,0
 mov bl,val1
 mov al,val2
 mul bl
 aam
```

bl	05
al	06
ax	001E
ax	0300

As it will update ax Should be ax

;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
```

	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	36						

Question 4 [10 Marks]

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

Code	Value

`.data`
`mFLAGS WORD 23997`
`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]`
`inc esi`
`mov dl, byte ptr [esi]`
`SUB dh, 48`
`SUB dl, 48`
`mov al, dh`
`mov dh, 10`
`mul dh`
`add dl, al`
 	
`MOV al, dl`
`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`
 	
`ror mDATA, 1`
 	
`skip:`
 	
`INVOKE ExitProcess, 0`
 	
`main ENDP`
`END main`
	Jump Taken/Not
	Not Taken
	
	DL
	1F (hex)
	OR
	31 (decimal)
	
	Demonstrate loop rough work
	
	mDATA
	6A (hex)
	
	mDATA
	B5 (hex)
	
	mDATA
	DA (hex)

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

```
;al*29 = al*(16+8+4+1)
;al*29 = al*16+al*8+al*4+al*1
mov al,2
mov ah,0
mov bx,0
add bl,al
mov cx,0
mov cl,al
shl cx,2      ;cx*4
add bx,cx
mov cx,0
mov cl,al
shl cx,3      ;cx*8
add bx,cx
mov cx,0
mov cl,al
shl cx,4     ;cx*16
add bx,cx
```

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
```

```
.386
.model flat, stdcall
.stack 4096
ExitProcess PROTO,
dwExitCode:DWORD
.data
     ary dq
0ABCDEF1234567890H,-
1,1929394959697900h,-2
     NumberOfOnes dw 0
.code
main Proc
mov ax,0
mov ecx,lengthof ary
mov eax,type ary
mov esi,0
```

```
L3:
   mov ebx,0
  L1:
   mov al,byte ptr ary[esi]
   mov edi,0
     L2:
        shl al,1
        jnc skipcount
        add NumberOfOnes,1
        skipcount:
        inc edi
        cmp edi,8
        jl L2
   inc ebx
   inc esi
   cmp ebx,type ary
   jl L1
LOOP L3
```

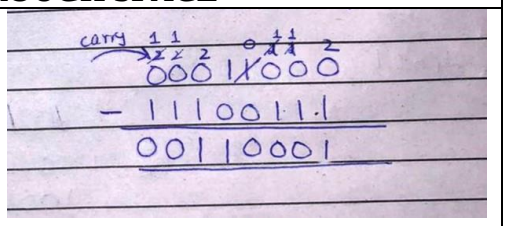
iii. Update Flags after executing following code and state which of the following jumps will be taken or not taken? Mention flags required for the jump. No marks for direct answer? Rough work is required.

CODE

```
mov al,24
mov bl,-25
cmp al,bl

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

ROUGH SPACE



	Taken	Not Taken	Flags Required
ja		✓	CF=0,ZF=0
jnb		✓	CF=0
jg	✓		SF=OF
jle		✓	SF!=OF
jnae	✓		CF=1

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

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
```

al	EAH 1110 1010
bl	CFH 1100 1111
al	25H 0010 0101
ah	00H 0000 0000
al	A5H 1010 0101
al	4BH 0100 1011
al	2CH 0010 1100
al	0Bh 0000 1011

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

ROUGH