National University
of Computer & Emerging Sciences Peshawar Campus

[illegible]

NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES

Student Name: Solution

Roll No: ____________

Program: ____________

Semester: Fall-2022

Time Allowed: 01 hour

Course: Computer Organization & Assembly Language

Examination: Sessional-II

Total Marks: 23 Weightage: 15

Date: 12th November, 2022

Instructor: Omar Bin Samin

NOTE: Attempt all questions.

NO ANSWER SHEET REQUIRED

Q1. Do as directed.

a. Show the values for AX and CF for the given program: (CLO: 3) (Marks: 2)

```
[org 0x100]
mov al, 10000011b
sar al,3
mov ax, 0x4C00
int 0x21
```

$\boxed{AX = 00F0}$ $\boxed{CF = 0}$

$1000\ 0011 \longrightarrow CF$

$\underbrace{1111}_{F}\ \underbrace{0000}_{0}$

b. Show the values for AX and CF for the given program: (CLO: 3) (Marks: 2)

```
[org 0x100]
mov al,10000011b
rcl al,2
mov ax, 0x4C00
int 0x21
```

$\boxed{AX = 000D}$ $\boxed{CF = 0}$

CF $\boxed{0} \leftarrow 1000\ 0011 \leftarrow$

$\boxed{0}$ $\underbrace{0000}_{0}\ \underbrace{1101}_{D}$

c. Show the values for AX and CF for the given program: (CLO: 3) (Marks: 2)

```
[org 0x0100]
mov ax, 0x4080
mov bx, 0x8080
shld ax,bx,1
mov ax, 0x4c00
int 0x21
```

$\boxed{AX = 8101}$ $\boxed{BX = 8080}$ $\boxed{CF = 0}$

$\boxed{\ }$ 1000 0000 1000 0000 BX

$\boxed{1} \leftarrow$ 0000 0001 0000 0000

$1 \leftarrow 0100\ 0000\ 1000\ 0000$ $\boxed{0} \leftarrow \boxed{1000\ 0001\ 0000\ 0001}$

d. How much segmented memory address space is required to represent Real Address Mode? (CLO: 2) (Marks: 2)

20-bit

e. What do we call the addresses generated by the CPU? (CLO: 1 & 2) (Marks: 2)

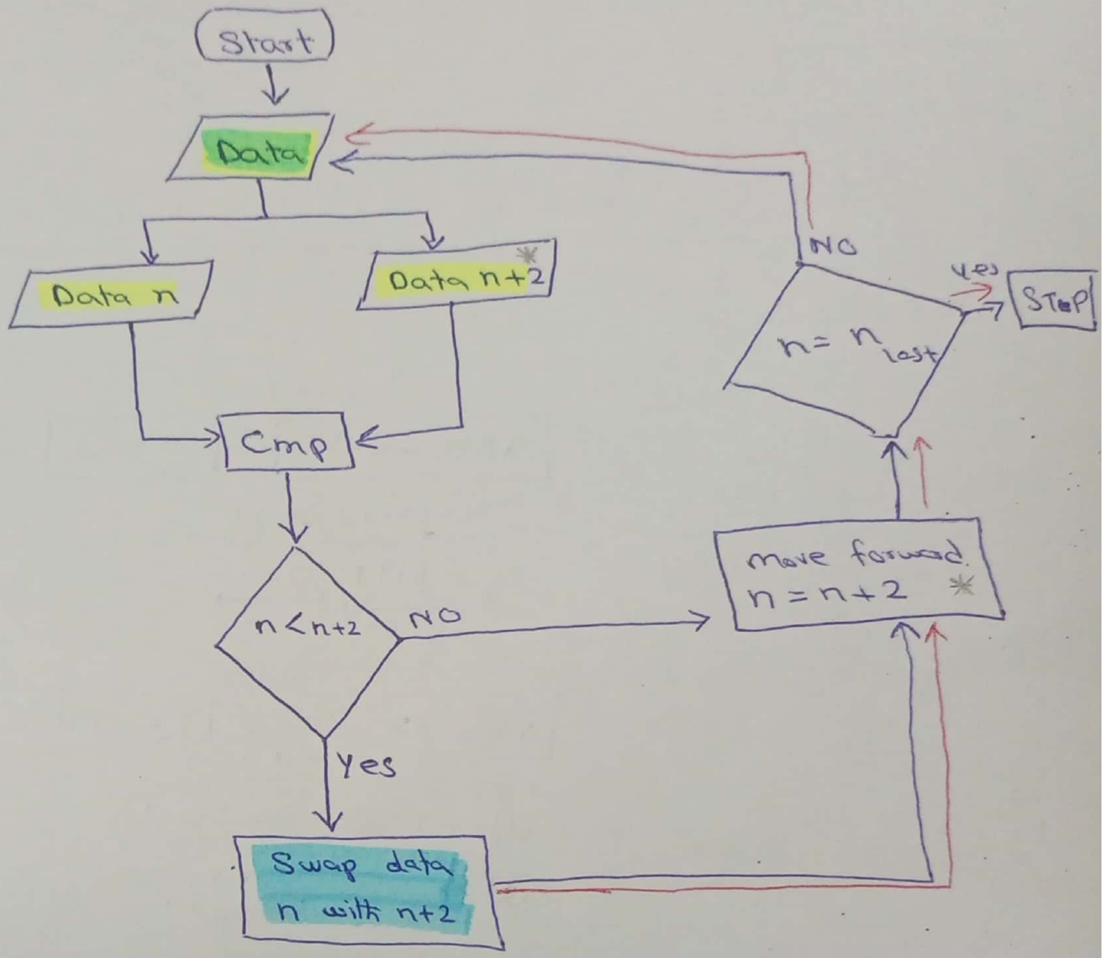
Logical / Virtual Address

Q2. Design an algorithm to sort the given data in descending order. (CLO: 4 & 5) (Marks: -

1 2 9 7 6

Note: Only flow chart required for the above given scenario. The used registers (AX, AH, AL, BX, BH, BL, CX, CH, CL, DX, DH, DL) for designing algorithm must be mentioned for every step.

Data = 1 2 9 7 6



Memory

AX → for Data

* BX → for incrementing by 2

→ cx → for loop

DX → for Data Swapping

Convert the given piece of code using conditional jump(s) such that the output remains same.
(CLO: 5 & 6) (Marks: 3)

```
[org 0x0100]
mov ax,1
mov bx,3
mov cx,5
tag1:
add ax,bx
cmp ax, 0x0D
loope tag1
mov ax,0x4c00
int 0x21
```

①

```
[org 0x100]
mov ax, 1
mov bx, 3
mov cx, 5
tag1:
 add ax, bx
cmp ax, 0x0D
je tag1
mov ax, 0x4c00
int 0x21
```

②

```
[org 0x100]
mov cx, 1
mov bx, 3
mov cx, 5
tag1:
add ax, bx
sub cx, 1
cmp cx, 0
je exit
cmp ax, 0x0D
je tag1
exit:
mov ax, 0x4c00
int 0x21
```

Q4. Show the contents of AX and BX along with all given flags for all executable instructions given below.
(CLO: 4 & 5) (Marks: 5)

Jump Instructions	Conditions
JG	ZF = 0 & SF = OF
JL	SF != OF

Instruction	AX	BX	OF	SF	ZF	PF
[org 0x100]						
mov al, 2	00 02	00 00	0	0	0	0
mov bx, 0xFFFF	00 02	FF FF	0	0	0	0
add ax, bx	00 01	FF FF	0	0	0	0
cmp ax, 1	00 01	FF FF	0	0	1	1
jp tag1	00 01	FF FF	0	0	1	1
jmp tag2						
tag1:						
sub al, bl	00 02	FF FF	0	0	0	0
mov bh, bl	00 02	FF FF	0	0	0	0
cmp al, bl	00 02	FF FF	0	0	0	1
jg tag2	00 02	FF FF	0	0	0	1
jl tag3						
mov ax, 0x4C00						
int 0x21						
tag2:						
sub ax, 2	00 00	FF FF	0	0	1	1
mov ax, 0x4C00						
int 0x21						
tag3:						
add al,0A1H						
mov ax, 0x4C00						
int 0x21						

[illegible]