Day: _____ Date: __/__/20__

Name: Mehammad Laraib Akhtar

Section: BCS-3B

Roll no: 21L-5294

Course: COAL

Assignment: 1

Q1 a. add ax, bx

Ax = 0x334A

Bx = 0x45F1

Cx = 0x8934

```
 334A
+45F1
 793B
```

ZF = 0 SF = 0

CF = 0 OF = 0

b. add cx, bx

```
 8934
+45F1
 CF25
```

ZF = 0 SF = 1

CF = 0 OF = 0

c. $45F1_{16}$

```
 45F1
-   6
 45EB
```

ZF = 0 SF = 0

CF = 0 OF = 0

Q2

a. 0xb900

big Endian format

lower address: b9 Higher address: 00

Day: Date: 1/20

b. 0x4567

lower address: 45

Higher address: 67

c. 0xAA99

lower address: AA

Higher address: 99

Question 3

a. FFFF:4312

FFFF0

+ 4312

10430 2

104302

b. 1DEF:0001

1DEF0

+ 0001

1DEF1

c. 14FF:1111

14FF0

+ 1111

16100

Day:

Date: / /

Q4. [org 0x0100]

```
mov ax, 10
mov bx, 5
mov cx, ax
mov ax, 0
l1: add ax, cx
sub bx, 1
cmp bx, 0
jne l1

mov ax, 0x4c00
int 0x21
```

Q5 a) bp-di

invalid : base register and index registe

addition possible only.

b) bp+si

0x220 + 0x0110

0220

+ 0110

0x0330

c) bx - 0x12

0034

- 0012

0x0022

d) bx+bp

invalid: cannot have two base register in one memory access.

e) bx+ip

invalid

no memory access can be performed through instruction pointer.

f) bx +di

0x0034

+ 0x1101

0x1135

Q6. (a) bx+si

22AA

+ FEEF

1,2199

12199

+ 45820

47989

wraparound: segment wraparound.

(b) 0x4700 + 0x4247 + 0x10.

= 4700
4247
+ 0010
0x8957
~~0x4588~~
+0x4582
0x E177

Physical address wraparound.

Q7. (a) mov, ip, bx

- Ip cannot be over written
- mov ax, bx.

(b) move byte bx, [ip].

- ip cannot be manually accessed.

(c) mov si, al

- Size mismatch.
- mov bl, al.

(d) mov ax, [bx+bp+100]

- bx + bp cannot be performed both are base registers.
- mov ax, [bx+si+100]

Question 8

OF → 0

SF → 1

CF → 0

PF → 0

Question 9

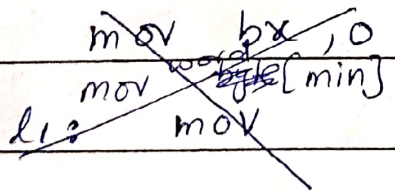
c)

There is no logical error in the code of Question 8.

Q10.

```
[org 0x0100]
mov al, [num1]
mov bl, [num1+1]
mov [num1], bl
mov [num1+1], al
mov al, [num1+2]
mov bl, [num1+3]
mov [num1+2], bl
mov [num1+3], al
mov ax, 0x4c00
int 0x21
num1: db 1,2,3,4.
```

Q11. [org 0x0100]




```
mov bx, 0;
mov ax, [array1]
mov [min], ax
l1:
add bx, 2
mov ax, [array1+bx]
cmp ax, [min]
```

```
jg l2
cmp bx, 10
jne l1
je end

l2:
mov [min], ax
Cmp bx 10
jne l1

end:
mov ax, 0x4c00
int 0x21

array1 : dw 5, 3, 8, 2, 5
min : dw 0
```