COMPUTER ARCHITECTURE (ET0738)

LAB TEST 2 (Practice)

Q1. Refer to the program given below. Find all the mistakes and write the corrections to the next column.

		Correction:
1	`org 100h`	
2	`jmp  again`	
3	`NumX  db 0`	
4	`NumY  db 1`	
5	`Sum   db 0`	
6	`start:`	
7	`move bx,0`	
8	`move cx,4`	
9	`mov al, NumX`	
10	`mov bl, NumY`	
11	`again:`	
12	`shl bl,2`	
13	`add ax,dx`	
14	`loop again`	
15	`mov [Sum],ax`	
16	`ret`	

a) Analyse the program, after the corrections compline and run. Describe what does this program do?
b) What are the memory addresses for variables NumX and NumY?
c) What is the value stored at variable "Sum"?
d) What is the value in BL register in each iteration?
e) If CX is set to 6, what will be the new summation result? Why?

Q2. Write an assembly program to sum up two 1x6 matrix arrays. Use the programming template below and complete the code. Summation result of SETA + SETB should be saved in SETB.

```
org 100h
jmp  start

SETA  db   1, 6, 1, 6, 7 ,2
SETB  db   4, 5, 8, 5, 2 ,6

start:

…
…




ret
```

a) What is the summation result?

Q3. **Challenge question**. Refer to Q2 and re-write the assembly code so that array SETA is summed with the reverse of array SETB.

```
org 100h
jmp  start

SETA  db   1, 6, 1, 6, 7 ,2
SETB  db   4, 5, 8, 5, 2 ,6

start:

…
…




ret
```

a) What is the summation result?

Solutions:

Q1.

		Correction:
1	org 100h	
2	jmp **start**	; jump to beginning of code which is at **start**
3	NumX db 0	
4	NumY db 1	
5	Sum **dw 0**	;saved result is word long
6	start:	
7	**mov bx,0**	;move should be mov
8	**mov cx,4**	
9	mov al, **[NumX]**	; receive a byte from the address of NumX
10	mov bl, **[NumY]**	; receive a byte from the address of NumY
11	again:	
12	shl bl,2	
13	add ax,**bx**	;dx never used so it must be bx
14	loop again	
15	mov [Sum],ax	
16	ret	

a) It will sum up a series of numbers 4,16, 64,0
b) NumX=0700:0102H, NumY=0700:0103H
c) 54h
d) 4h,10h,40h,0h
e) Sum is still 54h. SHL command shifts towards left two bits, after 3 iterations BL holds 0, shifting further doesn't change the value.

Q2.

```
org 100h
jmp  start

SETA  db   1, 6, 1, 6, 7 ,2
SETB  db   4, 5, 8, 5, 2 ,6

start:
lea  si,SETA
lea  di,SETB
mov  cx, 6
again:
mov  al, [si]
mov  bl, [di]
add  al,bl
mov  [di],al
inc  si
inc  di
loop again
ret
```

a) 05h,0Bh,09h,0Bh,09h,08h

Q3.

```
org 100h
jmp  start

SETA  db   1, 6, 1, 6, 7 ,2
SETB  db   4, 5, 8, 5, 2 ,6

start:
lea  si,SETA
lea  di,SETB
mov  cx, 6
again:
mov  al, [si]
mov  bl, [di+5]  ; Point to the end of array SetB
add  al,bl
mov  [si],al     ; Over write to array SetA to avoid conflict
inc  si
dec  di
loop again
ret
```

a) 07h,08h,06h,0Eh,0Ch,06h