



**Islamic University of Gaza**

**ECOM 2125 Assembly Language Lab**

**Eng. Eihab S. El-Radie**

**2014/2015**

**Date: May 13, 2015, Duration: 120 minutes**

## **Final Exam**

Student Name:	Solution								
Student Id:	1	2	0	0		-			

<b>Question</b>	<b>Points</b>
1	/15
2	/10
3	/20
4	/50
Bonus	/5
<b>Total</b>	<b>/100</b>

**(Q1) [15 Points] Write true or false for the following statements:**

- [ F ] 1. Int 16h and Int 10h are a MS-Dos-level access interrupts, while Int 21h is an Bios-level Access interrupt.
- [ F ] 2. The PROC directive begins a procedure and END directive ends a procedure.
- [ T ] 3. It is possible to call a procedure inside an existing procedure.
- [ T ] 4. Given the same task to accomplish, a nonrecursive procedure usually uses less memory than a recursive one.
- [ F ] 5. Function 01 Int 16h Check which key is pressed, and store the ASCII code in AL.
- [ T ] 6. LODSB, LODSW, and LODSD load a byte, word, or doubleword from memory at ESI into AL/AX/EAX, respectively.
- [ F ] 7. ESP is often called the base pointer or frame pointer because it holds the base address of the stack frame.
- [ T ] 8. The ROL (rotate left) instruction shifts each bit to the left. The highest bit is copied into the Carry flag and the lowest bit position.
- [ T ] 9. A subroutine's stack frame always contains the caller's return address and the subroutine's local variables.
- [ F ] 10. When a macro is invoked, the CALL and RET instructions are automatically inserted into the assembled program.
- [ F ] 11. The PUSH instruction cannot have an immediate operand.
- [ F ] 12. The JG instruction is used when comparing unsigned integers.
- [ T ] 13. No flags are affected by the NOT instruction.
- [ T ] 14. The TEST instruction does not alter the destination operand.
- [ T ] 15. The JO instruction is used after an operation involving signed integers.

**(Q2) [10 Points] Choose the correct answer:**

1. What is the content of Stack Pointer?

a. Address of the current instruction	b. Address of the next instruction
<b>c. Address of the top element of the stack</b>	d. None of the above

2. The content of AH register after the following operation is:

```
mov ax, 95h
mov bl, 10h
div bl
```

<b>(A). 05h</b>	(B). 09h
(C). 00h	(D). none

3. Which index register is used by the STOSD instruction?

(A). esp	(B). esi
(C). ebp	<b>(D). edi</b>

4. Suppose we want to convert the value 6Fh in AL to the value F6h. Which of the following instructions will do this:

(A). rol al,12	(B). rol al,4
(C). ror al,4	<b>(D). all the previous</b>

5. What will be the hexadecimal value of AX after the following instructions have executed?

```
mov ax, 6009h
stc
rcr ax, 1
```

<b>(A). B004h</b>	(B). 3004h
(C). 9600h	(D). 0096h

6. What will be the hexadecimal value of DX after the following instructions have executed?

```
mov dx, 0095h
clc
rcl dx, 1
```

(A). 004Ah	<b>(B). 012Ah</b>
(C). 008Ah	(D). 012Bh

### Example 1

```
1: .data
2: str1 BYTE "1324A2342424",0
3: .code
4: mov edi,OFFSET str1
5: mov al,'A'
6: cld
7: repne scasb
8: mov bl,[edi]
```

7. In Example 1, assume that `str1` is located at offset 00040010h. What will be the value of EDI after line 7 executes?

a. 00040011h	b. 00040013h
c. 00040014h	d. 00040015h

8. In Example 1, if we change line 6 to "std", what value will be moved to BL after line 8 executes?

a. ASCII code of "4"	b. ASCII code of "A"
c. ASCII code of "2"	d. cannot be determined

9. In Example 1, if we change line 7 to "repe scasb", what value will be moved to BL after line 8 executes?

a. ASCII code of "3"	b. ASCII code of "1"
c. ASCII code of "A"	d. ASCII code of "2"

10. The content of AH register after the following operation is:

```
mov ax,95h
mov bl,10h
div bl
```

a. 09h	b. 05h
c. 00h	d. none

### (Q3) [20 Points]: Answer the following Questions:

1. In the following instruction sequence, show the changed values of AL where indicated, in hexadecimal:

```
mov al,9Ch
not al           ; a. 63h
mov al,4Bh
and al,74h      ; b. 40h
mov al,86h
or al,42h       ; c. C6h
mov al,72h
xor al,0A5h     ; d. D7h
```

2. In the following instruction sequence, show the values of the Carry, Zero, and Sign flags where indicated:

```
mov al,00110011b
test al,2           ; a. CF= 0  ZF= 0  SF= 0
mov al,6
cmp al,5           ; b. CF= 0  ZF= 0  SF= 0
mov al,5
cmp al,7           ; c. CF= 1  ZF= 0  SF= 1
```

3. What is the binary value of AL after the following instructions have executed?

```
mov al,10000101b
clc
rcr al,1
```

**Ans:**

**0100 0010**

4. What is the binary value of AX after the following instructions have executed?

```
mov ax,0000000010011101b
mov bx,1010101010000000b
shld ax,bx,1
```

**Ans:**

**0000 0001 0011 1011**

5. Write a series of instructions that will multiply EAX by 18, using a combination of shift, MOV, and ADD instructions.

**Ans:**

```
mov ebx,eax ; make copy
shl eax,4   ; eax * 16
shl ebx,1   ; ebx * 2
add eax,ebx ; answer
```

6. Implement the following expression in assembly language.  
**ax = word1 \* (word3 % word2)**

**Ans.:**

```
mov ax,word3
mov dx,0
div word2
mov cx,dx
mov ax,word1
mul cx
```

7. If SS=0200h and SP=00FEh. What will be the value of SP, AX and EBX after executing the following code?

```
POP AX
POP EBX
```

Data stored in the stack segment is shown in the table below:

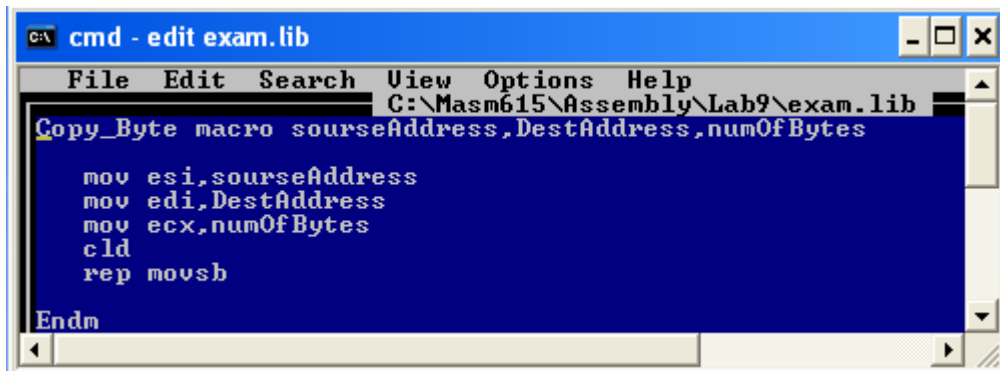
Value of SP	Value stored
108h	8Bh
107h	54h
106h	8Ch
105h	8Ah
104h	78h
103h	5Bh
102h	2Ah
101h	12h
100h	8Ch
0FFh	5Bh
0FEh	11h

```
SP=104h
AX=5B11h
BX=5B2A128Ch
```

**(Q4) Programming [45 Points]**

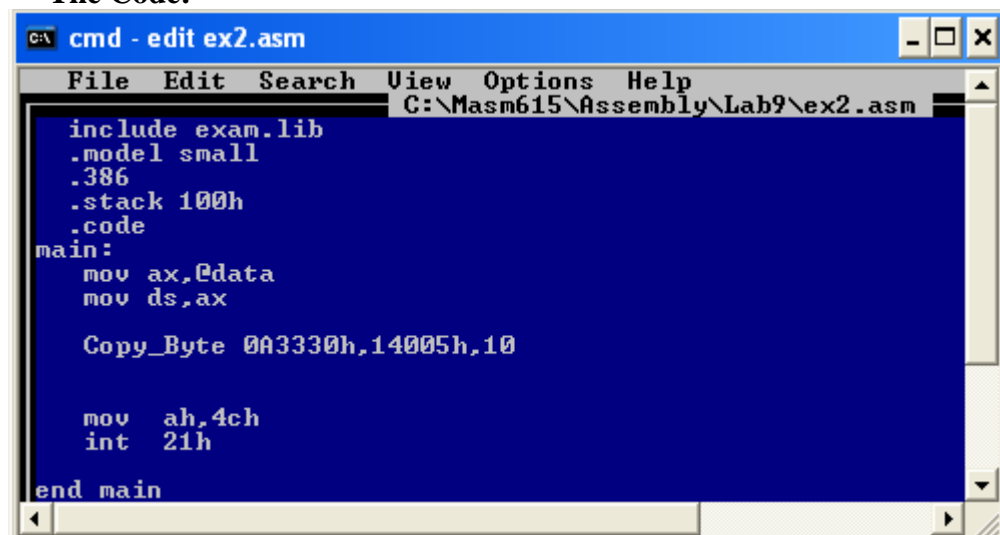
- A. [10 Points] Write a macro code named *Copy\_Bytes* taking three arguments *sourceAddress*, *DestAddress*, *numOfBytes*. This macro copy the first ten bytes from memory at address A3330h to address 14005h. Save this macro in a file named *exam.lib*, include it in your code and then invoke the macro.

**exam.lib:**



```
cmd - edit exam.lib
File Edit Search View Options Help
C:\Masm615\Assembly\Lab9\exam.lib
Copy_Byte macro sourceAddress, DestAddress, numOfBytes
    mov esi, sourceAddress
    mov edi, DestAddress
    mov ecx, numOfBytes
    cld
    rep movsb
Endm
```

**The Code:**



```
cmd - edit ex2.asm
File Edit Search View Options Help
C:\Masm615\Assembly\Lab9\ex2.asm
include exam.lib
.model small
.386
.stack 100h
.code
main:
    mov ax, @data
    mov ds, ax

    Copy_Byte 0A3330h, 14005h, 10

    mov ah, 4ch
    int 21h

end main
```

B. [15 Points] Write an assembly procedure that emulates the sumMid series:

SumMid (n) = n+sumMid (n/2)            where sumMid (I) =0 where I  
<=0

```
static int sumMid (int n){
    If (n<=0) return 0;
    else return n+sumMid(n/2);
}
```

**Solution:**

```
.model small
.386
.stack 180h
.data
.code
main:
mov ax,@data
mov ds,ax

mov eax,8
push eax
call sumMid
mov ah,4ch
int 21h                           ;terminate the programe
sumMid PROC
    push ebp
    mov ebp,esp
    mov eax,[ebp+6]               ; get n
    cmp eax,0                    ; n > 0?
    ja L1                        ; yes: continue
    mov eax,0                    ; no: return 0
    jmp L2
L1:  shr eax,1                    ;get n/2
    push eax                     ; sumMid(n/2)
    call sumMid
Return:
    mov ebx,[ebp+6]               ; get n
    add eax,ebx                   ; eax = eax + ebx
L2:  pop ebp                     ; return EAX
    ret 4                        ; clean up stack
sumMid ENDP
end main
```



- C. [20 Points] Write an assembly program that takes a sequence of numbers from the user, the program continues to ask for new numbers until the accumulative product of these numbers is greater than 1000 (base 10); if the user enters a non-digit character the program shall skip it.

**Solution:**

```
.model small
.386
.stack 180h
.data
messg1 db "Enter the sequense :",0dh,0ah, "$"
messg2 db 0ah,0d,"the result is greater than 1000.$"
result dw 1
limit = 1000
.code
main:
    mov ax,@data
    mov ds,ax
    mov ah,9
    mov dx,OFFSET messg1
    int 21h

l1:
    mov ah,8
    int 21h
    sub al,30h           ;convert Ascii to int
    cmp al,9           ;check if it is a number
    ja l1              ;contenue the loop if not digit
    mov ah,2           ;show the digit
    mov dl,al
    add dl,30h         ;convert int to Ascii
    int 21h
    sub al,30h         ;again convert Ascii to int
    movzx ax,al
    mul result
    mov result,ax
    cmp result, limit
    jb l1
    mov ah,9
    mov dx,OFFSET messg2
    int 21h
    mov ah,1           ;stop the execution
    int 21h
    mov ah,4ch
    int 21h

end main
```