

# National University of Computer and Emerging Sciences, Lahore Campus



Course:	Computer Organization and Assembly Language	Course Code:	EE213
Program:	BS(Computer Science)	Semester:	Fall 2019
Duration:	60 Minutes	Total Marks:	25
Paper Date:	25-Sep-2019	Page(s):	5
Section:	ALL	Section:	
Exam:	Midterm Exam 1	Roll No:	

**Instruction/Notes:** Answer in the space provided  
 You can ask for rough sheets but they will not be graded or marked  
 In case of confusion or ambiguity make a reasonable assumption.  
 Questions are not allowed  
 This is open book and open notes paper

Good luck!

Q1: (4 Points) Write a code for extended AND operation of 64 bit numbers *num1* and *num2* and store in *result*. You your answer in space provided. Note that your code should work for any values of *num1* and *num2*, not just for the values given below.

```
[org 0x0100]
jmp Start
; data is defined here
num1: dq 01020304h ; dq means define Quadword, it allocate 8 bytes
num2: dq 0A0B0C0Dh ;
result: dq 0 ; initially result is zero
Start:
```

; write a code to take **and** of *num1* and *num2* and store the result in *result*

```
mov ax, [num1]
mov bx, [num2]
mov [result], bx
and [result], ax

mov ax, [num1+2]
mov bx, [num1+2]
mov [result+2], bx
and [result+2], ax

mov ax, [num1+4]
mov bx, [num1+4]
mov [result+4], bx
and [result+4], ax
```

```
mov ax, [num1+6]
mov bx, [num1+6]
mov [result+6], bx
and [result+6], ax
```

```
; your code ends here
mov ax, 0x4c00
int 0x21
```

Roll Number: \_\_\_\_\_

Q 2 (10 points). Given an array of words with last element as -1, write a code to delete the even numbers from array. Example given below. (you can assume that -1 will not be in the array only at end)  
 Array: dw 10, 13, 96, 16, 18, 51, 88, 45, 2, 4, 3, -1  
 After your code finishes executing Array should be as follow.

13, 51, 45, -1, -1, -1, -1, -1, -1, -1, -1,

All the odd numbers are at start of Array in same order as were in given and even numbers are replaced by -1 and placed at the end of Array

NOTE:: YOU CANNOT DECLARE ANY OTHER ARRAY FOR WORKING AND YOUR CODE SHOULD WORK WITH ARRAYS OF ALL SIZES

```
[org 0x0100]
jmp start
; data is defined here
Array: dw 10, 13, 96, 16, 18, 51, 88, 45, 2, 4, 3, -1
start:
```

; write a code here using a loop

```
mov bx, 0
```

loop1:

```
mov ax, [result+bx]
```

```
cmp ax, -1
```

```
jz exitloop1
```

```
add bx, 2
```

```
test ax, 1
```

```
jnz loop1
```

```
mov [result+bx-2], -2
```

```
jmp loop1
```

exitloop1:

```
mov bx, 0
```

```
mov cl, 0
```

loop2:

```
mov ax, [result+bx]
```

```
cmp ax, -1
```

```
jz endloop2
```

; your code ends here

```
mov ax, 0x4c00
```

```
int 0x21
```

This loop checks if the last bit of a number is 1 (it is odd) then it jumps to the start. If it is zero then (it is even), it replaces it with -2. When the number reached is -1, then it exits the loop.

This loop checks, if the number is -1, then end the loop. If the number is -2 then it increments cl. And lastly if the number is not -1 and -2



space for Q2

```

cmp ax, -2
jnz next
add bx, 2
inc cl
jmp loop2
    
```

```

next:
add bx, 2
cmp cl, 0
    
```

```

jz loop2
mov si, 0
    
```

loop3:

```

cmp [result+si], -2
    
```

```

jz outloop3
    
```

```

add si, 2
    
```

```

jmp loop3
    
```

outloop3:

```

mov [result+si], ax
    
```

```

mov [result+bx-2], -2
    
```

```

jmp loop2
    
```

end loop2:

```

mov bx, 0
    
```

loop4:

```

mov ax, [result+bx]
    
```

```

cmp ax, -1
    
```

```

jz terminate program
    
```

```

add bx, 2
    
```

```

cmp ax, -2
    
```

```

jnz loop4
    
```

```

mov [result+bx-2], -1
    
```

```

jmp loop4
    
```

terminate program:

School of Computer Science

```

mov ax, 0x4C00
    
```

```

int 0x21
    
```

then it first checks if  $cl > 0$ , if it is not then it move forward, otherwise it ~~transverse~~ reverse from the start and put the value at first -2 and on its original space it puts a -2 and moves forward.

5+5

This loop puts a -1 at every place where a -2 is stored

Bytes →	0	1	2	3	4	5	6	7
DS:0100	F9	1E	BD	7A	CC	6A	C4	14

num: dq 14C46ACC7ABD1EF9h ; dq means define Quadword, it allocate 8 bytes

IV. (2 points) How following label num will be stored in memory, let num is stored at 0100 offset.

<p>mov al, 05h ; h is for hex test al, 09h</p> <p>What is the value of AL at the end of this code?</p> <p><math>ax = 05h</math></p> <p>What is value of ZF at the end of this code?</p> <p><math>ZF = 0</math></p>	<p>III. (2 points)</p>
--	------------------------

<p>mov ah, 0 mov al, 172 mov bl, 128 add al, bl</p> <p>What is the values of AX at the end of this code?</p> <p><math>AX = 2C</math></p>	<p>II. (1 points)</p>
--	-----------------------

<p>mov al, -128 mov bl, 117 cmp bl, 117 ja exit</p> <p>What is the value of OF, CF and SF at the end of following code?</p> <p><math>CF = 0</math> <math>SF = 0</math> <math>OF = 1</math></p> <p>Is the jump taken or not?</p> <p>Yes</p> <p>0000 1010 - 0111 0101 ----- 0111 1111</p>	<p>I. (2 points)</p>
---	----------------------

answers.

03 (7 points) Execute following codes and answer the questions given in second column. Give reason and show working in



(4 points)

Given that: CS = 5645h, DS = 1000h, ES = 6783h, SS = FFFFh, BX = 4567h, SI = FFFFh, DI = 2000h, BP = 4700h, SP = 4500h

Write the physical address of the memory locations. (Show your working)

Memory Location	Physical Address in hex
1 (2 points) [CS:bx + di]	5C9B7
11 (2 points) [bp + si + 10]	14709

$$\begin{array}{r} 4567 \\ + 2000 \\ \hline 6567 \end{array}$$

bx+di

$$\begin{array}{r} 56450 \\ + 6567 \\ \hline 5C9B7 \end{array}$$

CS:bx+di

$$\begin{array}{r} 4700 \\ + FFF \\ \hline 4709 \end{array}$$

BP+SI+10

excluded 4709

$$\begin{array}{r} 10000 \\ + 4709 \\ \hline 14709 \end{array}$$

DS:BP+SI+10

Section: