COLLEGE OF COMPUTING AND INFORMATION SCIENCES

Final-Term Assessment Fall 2020 Semester

Class Id	105753/104961	**Course Title**	Computer Organization and Assembly Language
Program	BSCS	**Campus / Shift**	Main Campus / Morning
Date	13 December 2020	**Total Marks**	100
Duration	03 hours	**Faculty Name**	Dr. Muhammad Taha Jilani/ Ms.Yumna Shahzad
Student Id	10187	**Student Name**	NIMRAJAMIL

Instructions:

- Filling out Student-ID and Student-Name and your Class-ID on exam header is mandatory.
- Do not remove or change any part of exam header or question paper.
- Write down your answers in given space or at the end of exam paper with proper title "Answer for Question# _ _".
- Handwritten text should be on A4 size page with clear visibility of contents.
- Before capturing the images, students must write their name/SID on the top of each page.
- Only PDF format is accepted (Student are advise to install necessary software)
- While uploading the answer script, rename it to, StudenID_StudentName.pdf
- In case of CHEATING, COPIED material or any unfair means would result in negative marking or ZERO.
- A mandatory recorded viva session will be conducted to ascertain the quality of answer scripts where deemed necessary.
- **Caution:** Duration to perform Final-Term Assessment is **03 hours only**. Extra 01 hours are given to cater all kinds of odds in submission of Answer-sheet. **Therefore, if you failed to upload answer sheet on LMS (in PDF format) within 03 hours limit, you would be considered as ABSENT/FAILED.**
- You need to attempt all questions
- For initial 30 minutes the instructor will remain available on Zoom to resolve any queries.

Q.1. Translate the following pseudocode into assembly language, assuming all values are unsigned. (10 marks)

```
if(( val1 >= val2 ) and (val2 < val3))
{
        while( val1 < 20 )
        {
         val1 = val1 + 1
         val3 = val3 * (val1 – val2) mod 20
        }
}
else if( val1 < val2 )   {
        val3 = 0
}
else
  val3 = 1;
```

(Restriction: Don’t use .if .while and .repeat directives of Irvine)

```
INCLUDE Irvine32.inc


.data
val1 dword ?
val2 dword ?
val3 dword ?
.code
Main proc
mov eax, val1
mov ebx, val2
mov ecx, val3
        cmp eax, ebx            ; FIRST CONDITION IN IF  : val1 >= val2
jbe blockelseif
cmp ecx, ebx            ;           AND SECOND CONDITION IN IF PARTval3 > val2
jb blockelseif
mov edx,20
top:        ;WHILE LOOP START
cmp eax, edx      ;VAL1<20
jae end           ;QUIT IF FALSE
        inc eax           ;IF CONDITION MET val1 = val1 + 1
        sub eax, ebx      ;val1 – val2
mov eax, eax
cdq
idiv 20            ; (val1 – val2) mod 20
mul ecx            ; val3 = val3 * (val1 – val2) mod 20
        Mov eax,ecx
        Jmp top
blockelseif:
```

```
cmp ebx, ecx            ;VAL1<VAL2
jbe blockelse    ;IF NOT MET JUMP TO ELSE


mov ecx, 1              ;ECX=1
mov val3, ecx         ;VAL3=ECX

blockelse:
mov ecx, 0           ;ECX=0
mov val3, ecx


END_label:                    ; to jump out of loop

exit
main endp
End main
```

Q.2. Answer the following questions. (10 marks)

a) Consider the following code:

```
    mov ax, 0h
    mov cx, 0Ah
doLoop:
         dec ax
         loop doLoop
```

What is the value of the ax register after the completion of the doLoop?

Q2-A-ANSWER :

After executing these lines the value of AX = -10 because at initial stage we have AX=0 AND CX,10 it will keep on executing the loop till CX= 0.

b) Find the value of AX after executing the given instructions, while considering the DS = 0042, AX = FFFF, BX = 0100, and memory address 0042:0100 = EEEE.

ANSWER Q2-B

```
MOV AX, BX          ;AX=0100
MOV AX, [BX]        ;AX=EEEE or 1110111011101110
```

c) For the given assembly code, find how many times this loop will be executed? `mov ecx,0`

```
X2:
     inc ax
     loop X2
```

ANSWER Q2-C: Loop will execute : 4,294,967,296 times because 2^32 = 4,294,967,296 and when the loop execute first time it will decrease the value of ecx from 0 to –1 till 2^32

d) What will be move to ECX counter?

```
.data
arrayCount WORD 30 DUP(?),0,0
.code
mov ecx,SIZEOF arrayCount             ; ?
```

ANSWER Q2-D: ECX= SIZEOF ARRAY COUNT and sizeof = type*lengthof , sizeof = 4*30 =120,so ecx=120

Q.3. A hyper mart wants to calculate its daily sales, each sale is recorded into an array of data. Assuming that all data stored in an array with large number of elements, for example:

SalesArray 100h, 425h, 660h,...... 999h

You need to write an assembly code that can calculate the sum of this 16-bit integers array. Within the comment field, describe each instruction also. (10 marks)

ANSWER Q3:

```
.DATA
        SalesArray WORD 100h, 425h, 660h,...... 999h
.code
        MOV ESI,OFFSET SalesArray           ;esi = 100h
        Mov ecx,LENGTHOF SalesArray         ;ecx= length of the array/no of elements
        Sumofarray:                         ;loop for summing the array
        MOV AX,[ESI]                        ;ax=100h
        ADD ESI, TYPE SalesArray            ;esi=esi+2
        ADD AX,[ESI]                        ;AX=100H+425,
        Loop Sumofarray
```

Q.4. Do the following tasks using bitwise operations: (10 marks)

a. Write a single instruction using 16-bit operands that clears the high 8 bits of AX and does not change the low 8 bits.

ANSWER Q4 PART-A: AND AX,0000000011111111

b. Write a single instruction using 16-bit operands that sets the high 8 bits of AX and does not change the low 8 bits.

ANSWER Q4 PARTB: OR AX,1111111100000000

c. Write a single instruction (other than NOT) that reverses all the bits in EAX.

ANSWER Q4 PARTC: XOR EAX,11111111111111111111111111111111

d. Write a single instruction that can take complement of flag-status register.

ANSWER Q4 PARTD:

Q.5. (a) There are three decimal numbers, 4850, 3920, and 6675 that are stored as 32-bit hexadecimal integers in EAX, EBX, and ECX registers, respectively. You need to create a procedure named as “SumOfThree” that will calculate their sum and then returns the total sum into EAX register.

(Show calculation steps in comments field for each instruction)

ANSWER Q5-A:

```
.code
Mov eax,4850            ;moving given values in registers
Mov ebx,3920            ;moving given values in registers
Mov ecx,6675            ;moving given values in registers
Call sumofthree           ;calling procedure
Call writeint
Endp
End main
Sumofthree:            ;proc start
Add eax,ebx            ;adding 2 registers and storing the result in eax
Add eax,ecx            ; adding 2 registers and storing the result in eax
Ret                    ;returning the answer in eax register
Sumofthree Endp        ;end proc
```

(b) For the following declared data, you need to find the register values in hexadecimal for the given instructions

```
.data

DoubleVal DWORD A1B2C3D4h

WordsVal WORD AB12h, CD34h, EF56h

.code

mov ax, WORD PTR [DoubleVal+2]        ; ax = A1B2
mov ebx, DWORD PTR WordsVal
                                      ; ebx = CD34EF56h
```

Q.6. Write instructions that:

(i) Jump to label “Greater” when the signed integer in AX is greater than the integer in CX

ANSWER Q6-A:

```
CMP AX,CX
JG greater
```

(ii) Jump to label “L1” when the unsigned integer in DX is less than or equal to the integer in CX

ANSWER Q6-B:

```
CMP DX,CX
JBE L1
```

(iii) Jump to label “Lesser” when a 32-bit integer -42 is compared with 26

ANSWER Q6-C:

```
MOV AX,-42
MOV CX,26
CMP AX,CX
JA LESSER
```

Q.7. Convert the following expression into equivalent assembly code, assuming 32-bit integers:

```
Result = 10 + ( ( 10 - 5 ) * ( 8 - 6 ) ) / 2
```

(Note: Write down the calculation of each step in comments)

ANSWER Q7:

```
.DATA
.CODE
main proc
mov eax, 10      ;EAX=10
mov ebx, 5       ;EBX = 5
sub eax, ebx      ;EAX= 10-5=5
mov esi, eax      ;ESI=5
mov eax, 8        ;EAX=8
mov ebx, 6        ;EBX=6
sub eax, ebx       ;EAX=6-8=2
Mul esi            ;eax=esi*eax=5*2=10
Cdq                ;converting dword to quadword
mov ebx, 2        ;ebx=2
div ebx          ;10/2= 5
mov ebx, 10      ;ebx=10
add eax, ebx     ;5+10 = 15


call writedec
```

Q.8. You are supposed to write an assembly program in which a nested loop is created to perform the given sequence. A general purpose register should be incremented 50 times, while each time the content of another register are incremented for 10 times. Write the complete code to execute this program.

ANSWER Q8:

```
.DATA
.CODE
main PROC
   mov eax, 0
   mov ebx, 0
   mov ecx, 50       ;LOOP COUNTER
LOOP1:
       PUSH ECX
       mov ecx, 10
       inc ebx
 lOOP2:
       loop LOOP2
       POP ECX
       inc eax
       loop LOOP1
call writedec
call crlf
```

Q.9. Suppose an ATM machine that is installed within the bank, when you entered a numeric pin-code it will allow you to perform transaction and if the pin-code will be invalid it will reject the transaction. Keeping this into view, you are required to write a simple assembly code that will do, in a following manner:

- First, prompt with a message “Enter Your Pin-code:”
- Assume, the saved pin-code is already stored in register, for example EBX = 2020h, while user’s entered pin-code will be stored by a register, for example EAX = 2000h.
- Compare Pin-codes that are kept by both EAX and EBX registers
- If both are same, it should be jumped to ‘transaction’ lable.
- If both are not equal, it should jump to rejection, which will again prompt for “Enter Your Pin-code:” message.

ANSWER Q9:

```
INCLUDE Irvine32.inc
.data
BEFORETRANS byte "Enter Your Pin-code: ",0
UI1 DWORD ?
UI2 dword 2020h
AFTERTRANS byte "Transcation done",0
.code
main PROC
L1:
mov edx,offset BEFORETRANS
call writestring
call readint

mov UI1, eax
mov ebx,UI2

cmp eax,ebx
je T
loop L1
T:
mov edx,offset AFTERTRANS
call writestring
exit
main ENDP
END main
```

Q.10. You need to write appropriate instructions in assembly language that can perform the following calculations for the signed numbers. Also write the final content of the register(s) in hexadecimal and the status flag(s) in your answer.

(a)

Multiplicand	Multiplier
+6,123,321	-555

(b)

Dividend	Divisor
63	+5

[illegible] DEFINED IN IDIV INSTRUCTIONS.