FAST NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

KARACHI CAMPUS

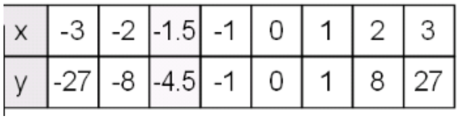
Quiz II Fall 2018

Section E & F

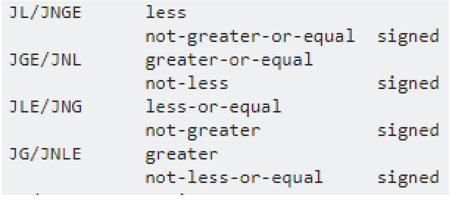
Computer Organization & Assembly Language (EE 213)

Student Roll No: ________________ Max. Marks: 2 Marks

Q1. Write x86 assembly snippet to find when both x and y goes in the positive quadrant.



x	-3	-2	-1.5	-1	0	1	2	3
y	-27	-8	-4.5	-1	0	1	8	27

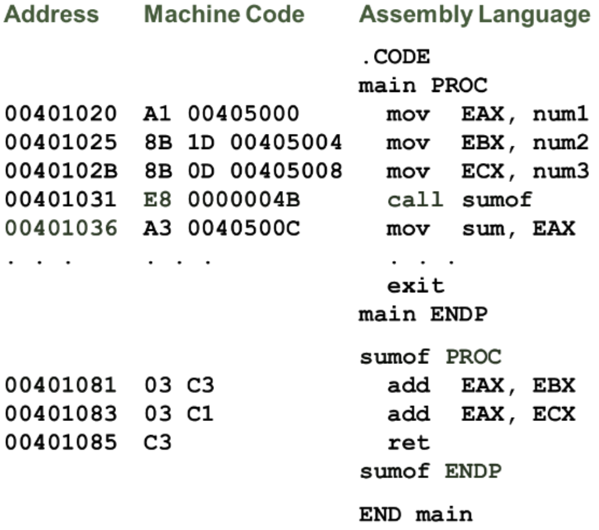


```
JL/JNGE     less
            not-greater-or-equal   signed
JGE/JNL     greater-or-equal
            not-less               signed
JLE/JNG     less-or-equal
            not-greater            signed
JG/JNLE     greater
            not-less-or-equal      signed
```

Q2. Write few x86 instructions to preserve eflags from changing while executing a CMP instruction.

Q3. Consider the following code. Suppose the sumof procedure starts with an address, which is the sum of address shown, and last four digits of your roll number as shown below. Recalculate the offset at address 00401031.

If your roll number is K19-**7856**. Then the number you add to the address should be 5678.



```
Address     Machine Code        Assembly Language
                                .CODE
                                main PROC
00401020    A1 00405000           mov   EAX, num1
00401025    8B 1D 00405004        mov   EBX, num2
0040102B    8B 0D 00405008        mov   ECX, num3
00401031    E8 0000004B           call  sumof
00401036    A3 0040500C           mov   sum, EAX
. . .       . . .                 . . .
                                  exit
                                main ENDP

                                sumof PROC
00401081    03 C3                 add   EAX, EBX
00401083    03 C1                 add   EAX, ECX
00401085    C3                    ret
                                sumof ENDP

                                END main
```

Q4. Suppose you receive an encryption key and a XOR encrypted string as follows:

```
str byte “oweiruemcmvnodfjpeporpelkjsldfkjoueoeldfjlsdfafsdsafdqwreqrwnbvnbvqwsa”,0
key byte 34
```

Write x86-assembly code to retrieve the original string by applying this key repeatedly to str. **Use indirect addressing mode and LOOP instruction should not appear in your code.**

Q5. BONUS QUESTION: Suppose EAX= F1AC6745h. Write x86 assembly code to process EAX such that EBX=4567ACF1h. That is last nibble of EAX becomes first nibble EBX, first nibble of EAX becomes last nibble of EBX and so on.