

48

1. For each of the below code segments give the name of the label where the next instruction after the compare statement will be executed. Assume for each problem that the contents of the registers are in their original state:

al: 0F2h

bl: 7Eh

cl: 80h

a) (5) cmp al, cl
jg here1
jmp here2

label: here1

b) (5) cmp al, cl
ja here1
jmp here2

label: here1

c) (5) cmp bl, cl
jge here1
jmp here2

label: here2

d) (5) cmp bl, cl
jb here1
jmp here2

label: here2

e) (5) cmp cl, bl
jnb here1
jmp here2

label: here2

2. (6) Give me three jump instructions which would follow unsigned integer comparisons (other than any listed above)

je, jne, jz

3. (6) Give me three jump instructions which could follow signed integer comparisons (other than those listed in number 1)

jz, jnz, jng

4. (5) Which conditional jump instruction is equivalent to the JNBE instruction?

ja ✓

5. (5) Which conditional jump instruction is equivalent to the JL instruction?

jnge ✓

6. (5) Write the instructions that jump to label L1 when the unsigned integer in DX is less than or equal to the integer in CX

cmp dx, cx
~~jle~~ L1
jbe

-4

7. (5) Write instructions that jump to label L2 when the signed integer in AX is greater than the integer in CX.

cmp ax, cx
~~ja~~ L2
jg

-4

8. (10) Write the assembly instructions that would follow the two that I have already written that would be equivalent to the following java statements (assume signed arithmetic):

if (iVal > iNum)
 iSum = iVal - iNum;
else
 iSum = iVal + iNum;

mov eax, iVal
mov ebx, iNum

cmp eax, ebx
ja subtract
jmp add

subtract:

sub eax, ebx
mov iSum, eax

add:

add eax, ebx
mov iSum, eax

jmp goaround

goaround:

9. (11) Write the assembly instructions that could be generated from the below java statements (assume signed arithmetic):

if ((iVal > iNum) && (iVal < 100))

iSum = iVal - iNum;

else

iSum = iVal + iNum;

mov eax, iVal
mov ebx, iNum

cmp eax, ebx
~~ja ANDI~~
jmp add

ANDI:
cmp eax, 100
~~jb subtract~~
jmp add

subtract:

sub eax, ebx
mov iSum, eax

add:

add eax, ebx
mov iSum, eax

done:

AND is an op
can't be
a Label

-2

-2

- jmp done

-6

-2

10. (11) Write the assembly instructions that could be generated from the below java statements (assume signed arithmetic):

if ((iVal > iNum) || (iVal < 100))

iSum = iVal - iNum;

else

iSum = iVal + iNum;

mov eax, iVal
mov ebx, iNum

cmp eax, ebx
~~ja subtract~~
~~jb OR~~

OR:

cmp eax, ebx
~~jb subtract~~
jmp add

subtract:

sub eax, ebx
mov iSum, eax

add:

add eax, ebx
mov iSum, eax

done:

no jmp necessary

100 -1

- jmp done -2

-7
-13

11. (11) Write a macro called **biggerVal** that would compare two dwords and would put the larger of the two values into a third dword.

```
macro biggerVal iVal1, iVal2, iVal3  
    cmp iVal1, iVal2  
    ja iVal1greater  
    jmp iVal2greater
```

Handwritten notes:
- *val3* (above *iVal3*)
- *local* (below *iVal1, iVal2*)
- *iVal1greater, iVal2greater, done* (above the jump instructions)
- *Can't do memory-to-memory* (with an arrow pointing to the *cmp* instruction)

iVal1greater:

mov iVal3, iVal1

jmp done

iVal2greater:

mov iVal3, iVal2

done;

end m

/10

/10
4