

Condition codes: new bits in hidden %eflags register.

Some instructions set those bits based on comparisons:

cmp, test

Other instructions change control flow (%eip) based on results:

jmp family

INSTRUCTION: **cmpl B, A**

computes A-B (but doesn't put result anywhere)

condition codes (incomplete):

zero flag : ZF=1 if (A-B) == 0 otherwise ZF=0

signed flag : SF=1 if (A-B) < 0 otherwise SF=0

INSTRUCTION: **jmp TARGET** always changes %eip to TARGET

INSTRUCTION: **je TARGET** %eip=TARGET if ZF==1

INSTRUCTION: **jne TARGET** %eip=TARGET if ZF== _____

INSTRUCTION: **jg TARGET** %eip=TARGET if _____

INSTRUCTION: **jge TARGET** %eip=TARGET if _____

INSTRUCTION: **jl TARGET** %eip=TARGET if _____

INSTRUCTION: **jle TARGET** %eip=TARGET if _____

Problem #6

Assume value of a is in %eax, and value of b is in %ebx

Write x86 assembly code for:

```
if (a > b) {  
    a++;  
}
```

Problem #7

Assume value of a is in %eax, and value of b is in %ebx

Write x86 assembly code for:

```
if (a > b) {  
    a++;  
} else {  
    b = a;  
}
```

Problem #8

Assume value of a is in %eax, and value of b is in %ebx

Write x86 assembly code for:

```
while (b > 0) {  
    a++;  
    b--;  
}
```