CS 271 Computer Architecture and Assembly Language
Self-Check for Lecture #6
Example Solutions (others possible)

Solve each problem using the following data segment:

```
.data
k           DWORD      ?
n           DWORD      ?
x           DWORD      ?
y           DWORD      ?
z           DWORD      ?
yes         BYTE       "Yes",0
no          BYTE       "No",0
maybe       BYTE       "Maybe",0
```

Assume that variables have been initialized. Write MASM code to implement the following high-level pseudo-code decision structures.

1.
```
if (k < n)
     print (yes);
else
     print (no);
```

2.
```
if (k < n)
     print (maybe);
else
     if (k > n)
          print (no);
          print (yes);
```

3.
```
if ((x < y) AND (y < z))
     print (yes);
else
     print (no);
```

4.
```
if ((x < y) OR (x > z))
     print (no);
else
     print (maybe);
```

NOTE: You cannot **cmp** memory to memory. At least one of the operands must be a register or a constant.

1.
```
      mov   eax, k
      cmp   eax, n
      jl    true1
      mov   edx, OFFSET no
      call  WriteString
      jmp   theEnd
true1:
      mov   edx, OFFSET yes
      call  WriteString
theEnd:
```

2.
```
      mov   eax, k
      cmp   eax, n
      jl    true_1
      jg    true_2
      mov   edx, OFFSET yes
      call  WriteString
      jmp   theEnd
true_1:
      mov   edx, OFFSET maybe
      call  WriteString
      jmp   theEnd
true_2:
      mov   edx, OFFSET no
      call  WriteString
theEnd:
```

3.
```
      mov   eax, x
      cmp   eax, y
      jge   false1
      mov   ebx, z
      cmp   y, ebx
      jge   false1
      mov   edx, OFFSET yes
      call  WriteString
      jmp   theEnd
false1:
      mov   edx, OFFSET no
      call  WriteString
theEnd:
```

4.
```
      mov   eax, x
      cmp   eax, y
      jl    true1
      cmp   eax, z
      jg    true1
      mov   edx, OFFSET maybe
      call  WriteString
      jmp   theEnd
true1:
      mov   edx, OFFSET no
      call  WriteString
theEnd:
```