

UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA

Facultad de Ingeniería, Arquitectura y
Diseño.

Ingeniería en Software y Tecnologías
Emergentes

Organización de Computadoras

Taller 8



Alejandro Palomares Ceseña

Ejercicio 1:

```
section .data
A dw 5
B dw 3
msg db "OK",10
len equ $-msg

section .text
global _start
_start:

    mov ax,[A]
    mov bx,[B]
    mov dx,ax
    sub dx,0
    jl negA

    mov dx,bx
    sub dx,0
    jl negB

    mov dx,ax
    sub dx,bx
    je iguales
    jg mayorA
    jl mayorB

negA:
jmp cont

negB:
jmp cont
31 jmp cont
32
33 cont:
34 mov dx,ax
35 sub dx,bx
36 je iguales
37 jg mayorA
38 jl mayorB
39
40 iguales:
41 jmp fin
42
43 mayorA:
44 jmp fin
45
46 mayorB:
47 jmp fin
48
49 fin:
50 mov eax,4
51 mov ebx,1
52 mov ecx,msg
53 mov edx,len
54 int 0x80
55 mov eax,1
56 mov ebx,0
57 int 0x80
```

Ejercicio 2:

```
section .data
N dw -4
msg db "OK",10
len equ $-msg

section .text
global _start
_start:

mov ax,[N]
mov dx,ax
sub dx,0
je cero
jl neg
jg pos

pos:
jmp fin
neg:
jmp fin
cero:
jmp fin

fin:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,len
int 0x80
mov eax,1
mov ebx,0
32 int 0x80
```

Ejercicio 3:

```
section .data
N dw 7
msg db "OK",10
len equ $-msg

section .text
global _start
_start:

mov ax,[N]
and ax,1
jp par
jnp impar

par:
jmp fin
impar:
jmp fin

fin:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,len
int 0x80
mov eax,1
mov ebx,0
int 0x80
```

Ejercicio 4:

```
section .data
A dw 30000
B dw 30000
msg db "OK",10
len equ $-msg

section .text
global _start
_start:

mov ax,[A]
add ax,[B]
jo ov
jno noov

ov:
jmp fin
noov:
jmp fin

fin:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,len
int 0x80
mov eax,1
mov ebx,0
int 0x80
```

Ejercicio 5:

```
section .data
A dw 65535
B dw 10
msg db "OK",10
len equ $-msg

section .text
global _start
_start:

    mov ax,[A]
    add ax,[B]
    jc car
    jnc nocar

car:
    jmp fin
nocar:
    jmp fin

fin:
    mov eax,4
    mov ebx,1
    mov ecx,msg
    mov edx,len
    int 0x80
    mov eax,1
    mov ebx,0
    int 0x80
```

Ejercicio 6:

```
section .data
A dw 5
B dw 9
C dw 2
msg db "OK",10
len equ $-msg

section .text
global _start
_start:

    mov dx,[A]
    mov ax,dx
    sub ax,[B]
    jl bmax
    jge c1

bmax:
    mov dx,[B]

c1:
    mov ax,dx
    sub ax,[C]
    jl cmax
    jge next

cmax:
    mov dx,[C]

next:
    mov si,[A]
    mov ax,si
    sub ax,[B]
    jg bmin
    jle c2

bmin:
    mov si,[B]

c2:
    mov ax,si
    sub ax,[C]
    jg cmin
    jle fin

cmin:
    mov si,[C]

fin:
    mov eax,4
    mov ebx,1
    mov ecx,msg
    mov edx,len
    int 0x80
    mov eax,1
    mov ebx,0
    int 0x80
```

Ejercicio 7:

```
section .data
A dw 9
B dw 2
msg db "OK",10
len equ $-msg

section .text
global _start
_start:

mov ax,[A]
mov bx,[B]
mov dx,ax
sub dx,bx
jle end
mov cx,ax
mov ax,bx
mov bx,cx

end:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,len
int 0x80
mov eax,1
mov ebx,0
int 0x80
```

Ejercicio 8:

```
section .data
msg db "OK",10
len equ $-msg

section .text
global _start
_start:

mov ax,0

loop:
inc ax
mov dx,ax
sub dx,10
jz fin
jmp loop

fin:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,len
int 0x80
mov eax,1
mov ebx,0
int 0x80
```
