

Nombre: Danice Martínez Bordes Grupo: 21

Nombre: \_\_\_\_\_

### Hoja de respuesta al Estudio Previo

1. Explica en detalle qué hacen las siguientes instrucciones. Si están sintácticamente mal escritas, indica el motivo:

movl \$1, %eax

Mueve el literal 1 al registro de 32 bits %eax

movl 1, %eax

Mueve el contenido de la dirección @1 de memoria al Registro %eax

movl \$1, eax

Mueve el literal 1 a la dirección de memoria referenciada por la etiqueta eax (ej. L1.1)

movl 1, eax

Mueve el contenido de la dirección de memoria @1 a la posición de memoria referenciada por eax

2. Dada la siguiente secuencia de instrucciones:

cmpl %eax, %ebx

jge fin

¿en qué condiciones se efectúa el salto?

Se ejecuta el salto si el resultado %ebx como entero es menor o igual que %eax (en docu): %ebx-%eax > 0

3. Dada la siguiente secuencia de instrucciones:

cmpl %ebx, %eax

jge fin

¿en qué condiciones se efectúa el salto?

Si el resultado de %eax - %ebx no tiene overflow ni detecta comprobación de signo

4. La traducción a código ensamblador del código C es:

movl \$0, %eax # i

for: cmpl \$100, %eax  
jge endfor

cmpl b, a

jge end-if

mult b, %ebx

subl a, %ebx

addl 1, %ebx

adde %ebx, a

inc %eax

jmp for

endif:

5. La traducción a código C del código ensamblador es:

```
while (i > 0){  
    if (a != b){  
        a = 3 * (b - a);  
    }--;
```

6. Explica cuál es la diferencia entre un *breakpoint* y un *watchpoint*.

Un *breakpoint* es una línea de programa donde quieres que la ejecución se detenga, en cambio un *watchpoint* es una condición que interrumpe y que permite detener

7. Explica cómo se puede hacer en *ddd* que un *breakpoint* sólo se pare cuando se ha ejecutado 15 veces.

Se debe colocar un *breakpoint* en la linea  
acceder a sus propiedades y para el ignore count a 15

8. Explica cómo se puede hacer en *ddd* para modificar el contenido de una dirección de memoria.

En *Data->Display*