

Nombre: RICHARD GRIJALDO

Grupo: 33

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`

MOV EL LONGWORD 1 AL REG. %eax.

`movl 1, %eax`

MOV EL LONGWORD de LA POSICIÓN de MEMORIA 1 AL REG. %eax.

`movl $1, eax`

AL REG. LI FALTA %.

`movl 1, eax`

AL REG. LI FALTA %.

2. Dada la siguiente secuencia de instrucciones:

`cmpl %eax, %ebx`

`jge fin`

¿en qué condiciones se efectúa el salto?

cmpl EA %ebx - %eax. SACA si %ebx ≥ %eax.

3. Dada la siguiente secuencia de instrucciones:

`cmpl %ebx, %eax`

`jge fin`

¿en qué condiciones se efectúa el salto?

SACA si %eax ≥ %ebx

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

```
movl $0, i
for: cmpl $300, i
     jge fi_for
     cmpl $a, $b
     jne fi_if
fi:  movl b, %eax
     subl a, %eax
     idivl 2
     sum %eax, a
```

```
fi_if: incl i
      jmp for
fi_for:
```

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

```
do {  
    if (a != b) {  
        b = b - a;  
        a = b;  
        a = a + a + b;  
    }  
    --i;  
} while (i != 0);
```

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

UN BREAKPOINT FA ATURAR EL PROGRAMA EN AQUELL PUNT.
UN WATCHPOINT FA ATURAR EL PROGRAMA QUAN ES COMPLEIX
UNA CONDICIÓN.

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

S'HA D'ANAR A LA CONFIGURACIÓ DELS BREAKPOINT I
CANVIAR EL IGNORE COUNT AL VALOR DESITJAT.

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

SI CLIQUEM A LA VARIABLE EN QÜESTIÓ I ANEM A
SET PODEM ASSIGNAR EL VALOR QUE VEGUEM