

Test 8: 10 puntos.

Escriba la opción correcta dentro de la casilla debajo de cada número de pregunta. Cada respuesta correcta vale $10/10 = 1,00$ puntos, 0 si no se contesta o está claramente tachada y $10/(3 \times 10) = -0,33$ si es errónea o no está claramente contestada. Se aconseja terminar de leer completamente cada pregunta antes de contestarla.

1	2	3	4	5	6	7	8	9	10

1. Se están probando varias implementaciones en ensamblador de la función factorial. ¿Cuál **no** debería funcionar?

```
unsigned f(unsigned n)
{
    if (n < 3)
        return n;
    else
        return n * f(n - 1);
}
```

- a)

```
unsigned total = 1;
asm("1: imul %1, %0 \n"
    "    loop 1b \n"
    :"+r"(total):"c"(n));
return total;
```
- b) Ninguna otra respuesta es correcta.
- c)

```
unsigned total = 1;
while (n)
    asm("imul %1, %0 \n"
        "dec %1 \n"
        :"+r"(total),"+r"(n));
return total;
```
- d)

```
unsigned total = 1;
while (n)
    asm("imul %1, %0 \n"
        "dec %1 \n"
        :"+r"(total):"r"(n));
return total;
```

2. ¿Qué registros podemos modificar a nuestro antojo en el interior de una función que siga el protocolo "System V AMD64 ABI", también llamados **salva invocante**?

- a) RAX, RCX, RDX, RDI, RSI, R8, R9, R10 y R11.
- b) Ninguna otra respuesta es correcta.

- c) RBX, RBP, R12, R13, R14 y R15.
- d) RSP.

3. ¿Dónde se pasan los primeros 6 parámetros a un función mediante el protocolo "System V AMD64 ABI"?

- a) En la pila.
- b) En la memoria.
- c) Ninguna otra respuesta es correcta.
- d) En registros.

4. ¿Cuál es la forma correcta de utilizar la instrucción "popcnt" para calcular el número de bits a 1 en un entero sin signo de 32 bits dentro de la siguiente función?

```
unsigned f(unsigned u)
{
    unsigned c;
    // ...
    return c;
}
```

- a)

```
asm("popcnt %1, %0" : "=r"(c) : "r"(u));
```
- b)

```
asm("popcnt %0, %1" : "=r"(c), "=r"(u));
```
- c) Ninguna otra respuesta es correcta.
- d)

```
asm("popcnt %0, %1" : "+r"(c) : "r"(u));
```

5. ¿A qué función puede corresponder el siguiente código ensamblador?

```
cmp    %esi,%edi
mov    %esi,%eax
cmovle %edi,%eax
ret
```

- a)

```
int f(int a, int b)
{
    if (a < b) return a;
```

```

        else        return b;
    }
b) int f(int a, int b)
{
    if (a == b) return a;
    else        return b;
}

```

c) Ninguna otra respuesta es correcta.

```

d) int f(int a, int b)
{
    if (a > b) return a;
    else        return b;
}

```

6. ¿A qué función puede corresponder el siguiente código ensamblador?

```

mov    %esi,%eax
ret

```

a) Ninguna otra respuesta es correcta.

```

b) int f(int a, int b)
{
    if (a == b) return a;
    else        return b;
}

```

```

c) int f(int a, int b)
{
    if (a < b) return a;
    else        return b;
}

```

```

d) int f(int a, int b)
{
    if (a > b) return a;
    else        return b;
}

```

7. ¿Qué registros estamos obligados a preservar en el interior de una función que sigue el protocolo

“System V AMD64 ABI”, también llamados **salva invocado**?

- a) RAX, RCX, RDX, RDI, RSI, R8, R9, R10 y R11.
- b) Ninguna otra respuesta es correcta.
- c) RBX, RBP, R12, R13, R14 y R15.
- d) RSP.

8. ¿En qué registro devuelven el resultado las funciones que siguen la convención de llamada a función “System V AMD64 ABI”?

- a) RCX b) RSP c) RAX d) RBX

9. ¿Cuál será el código de una función cuya definición es `int f(int a, int b, int c)` y que se implementa así?

```

mov    %esi,%eax
ret

```

- a) `return a;`
- b) Ninguna otra respuesta es correcta.
- c) `return c;`
- d) `return b;`

10. ¿Cuál será el código de una función cuya definición es `int f(int a, int b, int c)` y que se implementa así?

```

mov    %edi,%eax
ret

```

- a) `return c;`
- b) `return a;`
- c) Ninguna otra respuesta es correcta.
- d) `return b;`