



Nombre:

DNI:

Grupo:

Examen de Problemas (3.0 p)

1. **Acceso a arrays.** (0,4 puntos). Considerar el código C mostrado abajo, donde H y J son constantes declaradas mediante `#define`.

```
int array1[J][H];
int array2[H][J];

void copy_array(int x, int y) {
    array2[y][x] = array1[x][y];
}
```

Suponer que ese código C genera el siguiente código ensamblador IA-32:

```
copy_array:
    pushl %ebp
    movl %esp, %ebp
    movl 8(%ebp), %edx
    movl 12(%ebp), %eax
    leal 0(,%eax,8), %ecx
    subl %eax, %ecx
    addl %edx, %ecx
    leal (%edx,%edx,4), %edx
    leal (%edx,%eax), %eax
    movl array1(,%eax,4), %eax
    movl %eax, array2(,%ecx,4)
    popl %ebp
    ret
}
```

A la entrada:
8(%ebp) = x
12(%ebp) = y
#

constante	valor
H	
J	

¿Cuáles son los valores de H y J?

2. **Popcount.** (0,8 puntos). La práctica “popcount” debía calcular la suma de bits activados (pesos *Hamming*) de los elementos de un array. La siguiente función contiene un error (se han editado 2 líneas sobre la versión correcta) y produce resultados incorrectos.

```
int popcount4(unsigned* array,
              int len){
    int i,j;
    unsigned x;
    int val=0, result=0;

    for (i=0; i<len; i++){
        x = array[i];
        for (j=0; j<8; j++){
            val += x & 0x01010101;
            x >>= 1;
        }
        val += (val >> 16);
        val += (val >> 8);
        result+= val & 0xFF;
    }
    return result;
}
```

array	result	popcount
{0,1,2,3}		
{1,2,4,8}		
{1,16,256,1024}		
{5,4,3,2}		

Rellenar la tabla con los valores retornados por la función (columna **result**) y los valores que debieran haberse calculado (columna *popcount*) para cada uno de los cuatro arrays de test propuestos.

3. **Unidad de control** (0,3 puntos). Un procesador con una unidad de control microprogramada tiene una memoria de control de 300 palabras de 60 bits, de las que 200 son diferentes. ¿Qué ahorro en número de bits obtendríamos si usáramos nanoprogramación? Razone la respuesta con dos dibujos, uno sin nanoprogramación y otro con nanoprogramación.
4. **Entrada/Salida** (0,5 puntos). Disponemos de un microprocesador de 8 bits (bus de datos de 8 bits y bus de direcciones de 16 bits) con E/S independiente. Diseñar un sistema de E/S que permita acceder a los siguientes puertos: puerto 0x0440 de entrada y 0x0441 de salida. Utilizar lógica de decodificación distribuida. No emplear decodificadores.
5. **Diseño del sistema de memoria** (0,5 puntos). Disponemos de una CPU con buses de datos y direcciones de 16bit. Diseñar un sistema de memoria para la misma a partir de módulos SRAM de 16Kx4 y ROM de 8Kx8. La memoria ROM debe ocupar las direcciones 0x0000 a 0x7FFF y la SRAM 0x8000 a 0xFFFF. Se valorará la simplicidad del diseño. Dibujar también un mapa de memoria con las direcciones de inicio y final de cada módulo.
6. **Memoria cache** (0,5 puntos). Los parámetros que definen la memoria (direccionable por bytes) de un computador son los siguientes:
 - Tamaño de la memoria principal: 4 GB
 - Tamaño de la memoria cache: 1 MB
 - Tamaño de bloque: 64 BDetermine el tamaño de los distintos campos de una dirección en las siguientes condiciones:
 - a) Correspondencia completamente asociativa
 - b) Correspondencia directa
 - c) Correspondencia asociativa por conjuntos de 16 vías