

Nombre:**DNI:****Grupo:**

Test de Prácticas (4.0p)

Todas las preguntas son de elección simple sobre 4 alternativas.

Cada respuesta vale 4/20 si es correcta, 0 si está en blanco o claramente tachada, -4/60 si es errónea.

Anotar las respuestas (a, b, c o d) en la siguiente tabla.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

1. ¿Cuál es el valor mínimo (más negativo) que puede tomar un entero de 32bits en complemento a dos?

- a. -2^{32}
- b. $-2^{32} + 1$
- c. -2^{31}
- d. $-2^{31} + 1$

2. Después de ejecutar el siguiente código, ¿qué variables serán igual a 0? (Suponer ints de 32bits y longs de 64bits)

```
unsigned int a = 0xffffffff;  
unsigned int b = 1;  
unsigned int c = a + b;  
unsigned long d = a + b;  
unsigned long e =  
    (unsigned long)a + b;
```

- a. Ninguna
- b. c
- c. c y d
- d. c, d, y e

3. En la práctica 2 se pide sumar una lista de 32 enteros SIN signo de 32bits en una plataforma de 32bits sin perder precisión, esto es, evitando acarrees. ¿Cuál es el mínimo valor entero que repetido en toda la lista causaría acarreo con 32bits (sin signo)?

- a. 0xfc00 0000
- b. 0xfbff ffff
- c. 0x0800 0000
- d. 0x07ff ffff

4. En la práctica 2 se pide sumar una lista de 32 enteros CON signo de 32bits en una plataforma de 32bits sin perder precisión, esto es, evitando desbordamiento. ¿Cuál es el valor negativo más pequeño (en valor absoluto) que repetido en toda la lista causaría desbordamiento con 32bits (en complemento a 2)?

- a. 0xfc00 0000
- b. 0xfbff ffff
- c. 0xf800 0000
- d. 0xf800 0001

5. En la práctica 2 se pide calcular la media y resto de una lista de 32 enteros CON signo de 32bits en una plataforma de 32bits sin perder precisión, esto es, evitando desbordamiento. ¿Qué (media : resto) se debe obtener para una lista rellena a -1 salvo el primer elemento, que valiera -31?

- a. (-1 :-30)
- b. (-1 :-31)
- c. (-2 : 1)
- d. (-2 : 2)

6. ¿Cuál es el resultado de evaluar la expresión $1110_2 \wedge 1010_2$ en lenguaje C?

- a. 1111₂
- b. 1010₂
- c. 0110₂
- d. 0100₂

7. En la práctica 3 se pide calcular la suma de paridades de una lista de enteros sin signo. Suponer que un estudiante entrega la siguiente versión

```
int paridad5(unsigned* array,
             int len) {
    int i, k, result = 0;
    unsigned x;
    for (i = 0; i < len; i++) {
        x = array[i];
        for (k = 16; k == 1; k /= 2)
            x ^= x >> k;
        result += (x & 0x01);
    }
    return result;
}
```

Esta función:

- a. es correcta
 - b. falla para `array={0,1,2,3}`
 - c. falla para `array={1,2,3,4}`
 - d. no se puede marcar una y sólo una de las opciones anteriores
-
8. Suponer una memoria cache con las siguientes propiedades: Tamaño: 512 bytes. Política de reemplazo: LRU. Estado inicial: vacía (todas las líneas inválidas). Suponer que para la siguiente secuencia de direcciones enviadas a la cache: 0, 2, 4, 8, 16, 32, la tasa de acierto es 0.33. ¿Cuál es el tamaño de bloque de la cache?
- a. 4 bytes
 - b. 8 bytes
 - c. 16 bytes
 - d. Ninguno de los anteriores

9. Abajo se ofrece el listado de una función para multiplicar matrices $C = A \times B$.

```
void mult_matr(    float A[N][N],
                  float B[N][N], float C[N][N]){
    /* Se asume valor inicial C = {0,0,...} */
    int i,j,k;
    for (i=0; i<N; i++)
        for (j=0; j<N; j++)
            for (k=0; k<N; k++)
                C[i][j] += A[i][k] * B[k][j];
}
```

Suponer que:

- El computador tiene una cache de datos de 8 MB, 16-vías, líneas de 64 bytes.

- N es grande, una fila o columna no cabe completa en cache.
- El tamaño de los tipos de datos es como en IA32.
- El compilador optimiza el acceso a $C[i][j]$ en un registro.

Aproximadamente, ¿qué tasa de fallos se podría esperar de esta función para valores grandes de N?

- a. 1/16
- b. 1/8
- c. 1/4
- d. 1/2

-
10. Con los mismos supuestos, imaginar que se modifica la última sentencia (el cuerpo anidado) por esta otra

`C[i][j] += A[i][k] * B[j][k];`

de manera que se calcule $C = A \times B'$ (A por traspuesta de B). Aproximadamente, ¿qué tasa de fallos se podría esperar de esta nueva función para valores grandes de N?

- a. 1/16
- b. 1/8
- c. 1/4
- d. 1/2

-
11. La instrucción necesaria para cargar 0x07 en %eax es:

- a. `movb $0x07,%eax`
- b. `movl $0x07,%eax`
- c. `movb $0x07,%al`
- d. `movl $0x07,%ah`

-
12. La(s) instrucción(es) necesaria(s) para cargar el dividendo 0xa30bf18a en la pareja edx:eax como paso previo a una división sin signo son:

- a. `movl $0xf18a,%eax`
`movl $0xa30b,%edx`
- b. `movl $0xa30bf18a,%eax`
`xorl %edx,%edx`
- c. `movq $0xa30bf18a,%rax`
- d. `movl $0xa30bf18a,%eax`
`cld`

-
13. ¿Cuál es el efecto de la instrucción siguiente?

```
mov 8(%ebp),%ecx
```

- a. Suma 8 al contenido de ebp y almacena la suma en ecx
- b. Suma 8 al contenido de ebp, trata la suma como una dirección de memoria y almacena el contenido de esa dirección en ecx
- c. Suma 8 al contenido de la posición de memoria cuya dirección está almacenada en ebp, y almacena la suma en ecx
- d. Suma los contenidos de ebp y de la dirección de memoria 8 y almacena la suma en ecx

14. Si el registro eax contiene el siguiente valor binario:

31	2423	1615	8	7	0
11111111	10101010	01010101	11110000		

¿Cuál será el valor de %eax tras ejecutar la instrucción **xorb %al, %al**?

- a. 11111111 10101010 01010101 11110000
- b. 00000000 10101010 01010101 11110000
- c. 00000000 00000000 00000000 00000000
- d. 11111111 10101010 01010101 00000000

15. ¿Qué valor contendrá el registro edx tras ejecutar las dos instrucciones siguientes?

```
movl $-1, %edx  
movb $1, %dl
```

- a. 11111111 11111111 11111111 11111111
- b. 11111111 11111111 11111111 00000001
- c. 00000000 00000000 00000000 00000001
- d. 00000001 00000000 00000000 00000000

16. ¿Qué valor contendrá edx tras ejecutar las siguientes instrucciones?

```
xor %eax, %eax  
sub $1, %eax  
cld  
idiv %eax
```

- a. 0
- b. 1
- c. -1
- d. no puede saberse con los datos del enunciado

17. ¿Cuál de los siguientes fragmentos es correcto para comenzar un programa en ensamblador que conste de un solo archivo .s?

- a. **.text**
_start:
- b. **.section .text**
.local _start
_start:
- c. **.section .text**
.global _start
_start:
- d. **section .text**
.start _global
_start:

18. En un programa en ensamblador queremos crear espacio para una variable entera var inicializada a 1. La línea que hemos de escribir en la sección de datos es:

- a. **.int var 1**
- b. **var: .int 1**
- c. **.int: var 1**
- d. **int var 1**

19. ¿Cuántos operandos acompañan a la instrucción PUSH en IA32?

- a. 0
- b. 1
- c. 2
- d. 3

20. ¿Para qué se utiliza la función gettimeofday en la práctica de la "bomba digital"?

- a. Para cronometrar y poder comparar las duraciones de las distintas soluciones del programa.
- b. Para imprimir la hora en la pantalla.
- c. Para cifrar la contraseña en función de la hora actual.
- d. Para lanzar un error cuando el usuario tarde demasiado tiempo en introducir la contraseña o el PIN.