2º Grado Informática
Estructura de Computadores
13 de enero de 2020

Nombre:	
DNI:	**Grupo:**

Test de Prácticas (4.0p)

Todas las preguntas son de elección simple sobre 4 alternativas.
Cada respuesta vale 0.2p si es correcta, 0 si está en blanco o claramente tachada, -0.0$\widehat{6}$p si es errónea.
Anotar las respuestas (a, b, c ó 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. En la práctica "media" se pide como ejercicio previo sumar una lista de N enteros en doble precisión. Alguna de las siguientes técnicas no está relacionada con la aritmética en doble precisión:

a. acumulación de desbordamientos (OF, overflow flag)
b. acumulación de acarreos (CF, carry flag)
c. extensión con ceros
d. extensión de signo

---

2. En la misma práctica "media", el programa esqueleto ofrecido (suma.s) no es válido, en concreto... (marcar la opción FALSA)

a. no está preparado para sumar más de 9 elementos
b. no hace extensión con ceros de los elementos
c. no hace extensión de signo de los elementos
d. no consulta ni el flag de acarreo CF ni el de overflow OF

---

3. La práctica "popcount" debía calcular la suma de bits (peso Hamming) de los elementos de un array. Un estudiante entrega la siguiente versión de popcount5:

```
int popcount5(int* array, size_t len){
    size_t i,j;
    int x, val, result=0;
    for (i=0; i<len; i++){
      x = array[i];
      val = 0;
      for (j=0; j<8; j++){
        val += x & 0x01010101;
        x >>= 1;
      }
      val += (val >> 16);
      val += (val >>  8);
    result+=  val & 0xFF;
    }
    return result;
}
```

Esta función se diferencia de la versión "oficial" en los tipos de array y x.

Esta función popcount5:

a. produce siempre el resultado correcto
b. fallaría con array={0,1,2,3}
c. fallaría con array={0,-1,-2,-3}
d. no es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

---

4. Otro estudiante entrega la siguiente versión de popcount5:

```
int pcnt5(unsigned* array, size_t len){
    size_t i,j;
    unsigned x;
    int val, result=0;

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

Esta función sólo se diferencia de la versión "oficial" recomendada en clase en las condiciones del for interno.

Esta función popcount5 fallaría:

a. con array={1, 16, 256, 4096}
b. con array={1, 32, 1024, 32768}
c. con ambos ejemplos
d. con ninguno de los dos ejemplos

5. En una bomba como las estudiadas en prácticas, del tipo...

```
4006fa: lea    0x10(%rsp),%rbx
4006ff: mov    0x20096a(%rip),%rdx
400706: mov    $0x64,%esi
40070b: mov    %rbx,%rdi
40070e: callq  400570 <fgets@plt>
400713: mov    $0xd,%edx
400718: lea    0x200939(%rip),%rsi
40071f: mov    %rbx,%rdi
400722: callq  400560 <strncmp@plt>
400727: test   %eax,%eax
400729: je     400730 <main+0x51>
40072b: callq  400697 <boom>
400730: lea    0x1b5(%rip),%rsi
```

...la contraseña (alfanumérica) es...

a. el string almacenado a partir de 0x10(%rsp)
b. el string alm. a partir de 0x20096a+0x4006ff
c. el string almacenado a partir de 0x601058
d. el string alm. a partir de 0x200939+0x400718

6. En una bomba como las estudiadas en prácticas, del tipo...

```
400746: lea    0xc(%rsp),%rsi
40074b: lea    0x1ae(%rip),%rdi
400752: mov    $0x0,%eax
400757: callq  400590 <_scanf@plt>
40075c: mov    0x2008ee(%rip),%eax
400762: cmp    %eax,0xc(%rsp)
400766: je     40076d <main+0x8e>
400768: callq  400697 <boom>
40076d: callq  4006bb <defused>
```

...el código numérico (pin) es...

a. el entero almacenado a partir de 0xc(%rsp)
b. el entero alm. a partir de 0x1ae+0x40074b
c. el entero almacenado a partir de 0x6010560
d. el entero alm. a partir de 0x2008ee+0x400762

7. La función setup() de Arduino es llamada:

a. Al principio de cada iteración de la función loop()
b. Cuando se sube desde el entorno de desarrollo o se pulsa el botón de reset, pero no cuando se conecta la alimentación
c. Cuando se conecta la alimentación a la placa, se pulsa el botón de reset, o se sube desde el entorno de desarrollo
d. Cuando se conecta la alimentación a la placa, se pulsa el botón de reset, pero no cuando se sube desde el entorno de desarrollo

8. Suponga una memoria cache con las siguientes propiedades: Tamaño: 512 bytes. Política de reemplazo: LRU. Estado inicial: vacía (todas las líneas inválidas). Suponga que para la siguiente secuencia de direcciones enviadas a la cache: 0, 1, 2, 4, 8, 16, 32, 64, la tasa de acierto es 0,25. ¿Cuál es el tamaño de línea de la cache?

a. 4 bytes
b. 8 bytes
c. 16 bytes
d. Ninguno de los anteriores

9. En el programa size.cc de la práctica "cache", se accede al vector saltando...

a. de byte en byte
b. de 64 en 64 bytes
c. de 1 KB en 1 KB
d. de line en line bytes, donde line barre los valores desde 1 hasta 1K en un bucle for

10. En la práctica de cache hemos hecho una gráfica con el código size.cc ¿Qué forma tiene la gráfica que se debe obtener?

a. Forma de U (o V), con un tramo descendente y otro ascendente
b. Forma de /, una gráfica siempre creciente y sin escalones
c. Forma de media U seguida de \, es decir, un tramo descendente suave, un pequeño tramo horizontal, y un tramo descendente lineal
d. Una escalera con varios tramos horizontales

11. La línea de código ensamblador: **mov $msg, %rsi**

a. Copia en rsi los primeros 64 bits de memoria desde la posición apuntada por la etiqueta msg.
b. Copia en rsi todo el contenido de la cadena apuntada por msg.
c. Copia en rsi la dirección de memoria de 64 bits almacenada en memoria a partir de la posición indicada por la etiqueta msg.
d. Copia en rsi los 64 bits de la dirección msg.

12. En la práctica "media" se programa la suma de una lista de 16 enteros de 4 bytes para producir un resultado de 8 bytes, primero sin signo y luego con signo. Si la lista se rellena con el valor 0x0400 0000, ¿en qué se diferencian los resultados de ambos programas?

a. no se diferencian
b. en uno ocupa 32 bits, en otro 64 bits
c. en uno se interpreta como negativo, en otro como positivo
d. en uno los 32 bits superiores son 0xFFFF FFFF, en el otro no

---

13. ¿Cuál de las siguientes líneas declara un puntero a función en C?

a. int *func;
b. int func();
c. int *func();
d. int (*func)();

---

14. Si val es una variable de tipo unsigned long, entonces la sentencia: **val += (val >> 32);**

a. Pone siempre val a 0.
b. Deja siempre val al mismo valor que tuviera antes de la sentencia.
c. Su traducción incluye una instrucción shr seguida de una suma.
d. Su traducción incluye una instrucción sar seguida de una suma.

---

15. ¿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.

---

16. En la práctica de la bomba, el primer ejercicio consiste en ir saltando las “explosiones” mientras se depura el código, para lo cual se puede utilizar…

a. objdump o gdb
b. gdb o ddd
c. ddd o hexedit
d. hexedit u objdump

---

17. En la placa del kit de Arduino, las patillas de tierra vienen etiquetadas con la leyenda:

a. A0
b. 5V
c. GND
d. 3.3V

---

18. Las resistencias utilizadas en la práctica de Arduino

a. Son de color azul claro y tienen 5 bandas de color: las 3 primeras indican un valor, la 4ª banda es un multiplicador y la 5ª banda es la tolerancia
b. Son de color beige y tienen 4 bandas de color: las 2 primeras indican un valor, la 3ª banda es un multiplicador y la 4ª banda es la tolerancia
c. Tienen polaridad y el cátodo (polo negativo) es el extremo de la banda de color con una separación mayor respecto a las otras.
d. Tienen polaridad y el ánodo (polo positivo) es el extremo de la banda de color con una separación mayor respecto a las otras.

---

19. ¿Cuál de las siguientes afirmaciones sobre las caches es FALSA?

a. Casi ningún procesador actual tiene memoria cache L2.
b. Las direcciones a las que accede un programa no son completamente aleatorias, sino que se rigen por ciertos patrones de localidad.
c. Un procesador actual tiene varias caches de nivel 1.
d. La cache de nivel 3 no contiene toda la memoria que maneja el programa.

---

20. En el programa line.cc, si para cada tamaño de línea (line) recorremos una única vez el vector, la gráfica resultante es decreciente porque:

a. Cada vez que line aumenta al doble, el número de aciertos por localidad temporal aumenta, porque ya habíamos accedido a cada posición i del vector cuando lo recorrimos en el punto anterior del eje X.
b. Cada vez que line aumenta al doble, el número de aciertos por localidad espacial aumenta, porque ya habíamos accedido a cada posición i−1 del vector cuando lo recorrimos en el punto anterior del eje X.

c. Cada vez que line aumenta al doble, se accede con éxito a más posiciones del vector en niveles de la jerarquía de memoria más rápidos.
d. Cada vez que line aumenta al doble, realizamos la mitad de accesos al vector que para el valor anterior.