

ALUMNO \_\_\_\_\_

GRUPO \_\_\_\_\_

**CUESTIONES DE TEORÍA***(Puntuación: 1 punto. Las contestadas de forma errónea no descuentan)***1. ¿ Puede un programa escrito en lenguaje C ejecutarse sin contener la función main() ?**

- (a) Sí, siempre
- (b) No, nunca
- (c) Sí, pero sólo cuando contiene la cláusula #include
- (d) Sí, pero sólo cuando no contiene la cláusula #include

**2. ¿Cuál de las siguientes afirmaciones acerca de las capacidades de información es falsa ?**

- (a) 1 byte = 8 bits
- (b) 1 Kbyte = 1024 bytes
- (c) 1 Mbyte = 8 Mbits
- (d) 1 Gbit = 1024x1024bits

**3. ¿Cuál de los siguientes tipos de datos no puede considerarse como un tipo básico ?**

- (a) Entero

- (b) Real
- (c) Carácter
- (d) Vector

**4. ¿Cuál es el objetivo del scanf ?**

- (a) Visualizar el contenido de una variable por pantalla
- (b) Obtener el valor para una variable a través del teclado
- (c) Scanear un documento (abreviatura de "scan"+"file")
- (d) No es una instrucción del lenguaje C

**5. ¿Cuáles son los elementos básicos para controlar el orden de ejecución de las instrucciones de un programa ?**

- (a) Asignación, instrucción y sentencia
- (b) Secuencia, selección e iteración
- (c) Variable, constante, librería y función
- (d) Printf y scanf

**Cuestiones de Práctica:****1. (1 punto) Descubre al menos cinco errores que aparecen en el siguiente programa, cuyas líneas se han numerado para facilitar su identificación:**

```

1.  #include <stdio.h>
2.  #define TAM = 100
3.  #define C 5
4.
5.  struct estructura {
6.      int parcial;
7.      int total;
8.  };
9.
10. void main () {
11.
12.     int i;
13.     struct estructura v[TAM];
14.
15.     printf("Que constante quieres sumar a la sucesion?: ")
16.
17.     scanf("%d",&C);
18.
19.     for (i = 0; i <= TAM; i++) {
20.
21.         v.parcial[i] = i + C;
22.
23.         if (i > 0)
24.             v[i].total = v[i-1].total;
25.             v[i].total += v[i].parcial;
26.         else
27.             v[i].total = v[i].parcial;
28.
29.     }
30.
31.     printf("La suma vale %d\n", v[TAM-1].total);
32. }

```

2. (1 punto) Transforma el for que aparece a continuación en el while correspondiente.

```
for (exp1; exp2; exp3)
{
    instrucciones;
}
```

3. (1 punto) Indica que presenta en pantalla la ejecución de este programa:

```
#include <stdio.h>

void elevar(int a)
{
    a=a*a;
    printf("el valor de a es %d\n",a);
}

void main()
{
    int a;
    for (a=0;a<10;a=a+2)
    {
        elevar(a);
        print (" %d\n",a);
    }
}
```

## Problemas:

1. (1 punto) Crear la/s estructura capaz de almacenar la siguiente información sobre un cliente de un banco:

- Numero de cuenta (p.e. 00023534978626667800, 20 dígitos).
- Tipo de cuenta (A:ahorro, C:cuenta corriente, P:plazo fijo, V:cuenta vivienda).
- Saldo en euros.
- Fecha de apertura.
- Fecha de ultima actualización.
- Posee tarjeta de crédito (S/N)

2. (2 puntos) Definimos un vector de enteros como alternado si el valor de cada componente es de signo diferente al del anterior. Por ejemplo:

-2	3	-4	1	-90	2	...
----	---	----	---	-----	---	-----

6	-2	3	-1	2	-4	...
---	----	---	----	---	----	-----

Escribe un programa que lea desde teclado un vector de 100 enteros, compruebe si es alternado, y finalmente imprima por pantalla un mensaje adecuado indicando si es alternado o no.

3. (3 puntos) Se desea llevar el control de una gasolinera. En esta hay cinco surtidores diferentes: para gasolina super, normal, sin plomo, gas-oil y mezcla.

La estructura de datos para guardar la información necesaria para cada surtidor contendrá el nombre (cadena de caracteres) del combustible que contiene, una referencia (código numérico de tres cifras), cantidad de carburante que almacena y precio por litro. Tiene el siguiente aspecto:

```
struct surtidor {  
    char nombre[15];  
    int codigo;  
    float cantidad;  
    float precio;  
};
```

Se pide un programa que contenga una serie funciones encargadas de:

- Inicio. Asignar los valores iniciales a las estructuras de datos.
- Repostar. Un usuario indicará de que gasolina quiere (su referencia) y cuanto dinero desea gastarse. Si hay suficiente cantidad de combustible el sistema actualizará la cantidad que queda en el surtidor y dirá al usuario cuanto combustible se ha servido.

La función main de este programa contendrá la declaración del vector de estructuras que almacenará la información de los cinco surtidores y un menú para poder elegir entre las opciones de asignar valores iniciales, repostar ó salir del programa.