

# Fundamentos de Informática (Ing. Técnica Industrial). Examen Teórico. 2-7-2001

ESPECIALIDAD:  Electrónica  
GRUPO :  Mañana (A)

Electricidad  
 Mañana (C)

Mecánica  
 Tarde (B)

NOMBRE del ALUMNO: \_\_\_\_\_

1. Contestar a (1.5 p.):

- Explique el proceso que realiza la CPU cuando lee y ejecuta una instrucción.
- Calcule el valor binario de los siguientes valores hexadecimales: CF E50
- Determine los valores en base decimal de los siguientes números binarios: 0111 11010

2. Escriba la salida que produciría por pantalla la ejecución de este programa y explique brevemente su funcionamiento (1.5 p.):

```
#include <stdio.h>
void f1(int a, int *b, int *c){
    a++;
    *b=a*a;
    *c=a*a*a;
}
void main() {
    int x=2,y=3,z=4;
    f1(y,&x,&z);
    printf ("%d\t%d\t%d\n", x,y,z);
    f1(x,&y,&z);
    printf ("%d\t%d\t%d\n", x,y,z);
}
```

¿Qué utilidad tiene pasar un puntero a una variable como parámetro en una función?

3. Dadas las siguientes declaraciones:

```
struct fecha{
    int dia, mes, anio;
};

struct persona{
    char nombre[50];
    struct fecha fecha_nac;
    double altura;
    double peso;
};
```

- Implemente una función `escribir()` que muestre por pantalla todos los datos de todas las personas que hay almacenadas en un array de `struct persona`. La función debe tener **dos argumentos**, el primero un array de `struct persona` y el segundo un entero que indique el número de datos que hay en el array. La función no devuelve nada (0.5 p.).
- Escriba una función `leer()` para pedir todos los datos de una persona concreta. La función no tendrá argumentos y devolverá un `struct persona` con los datos leídos (0.5 p.).
- Suponga que tenemos la siguiente declaración de variables: `struct persona Equipo[100];`  
Indique si las siguientes llamadas a la función implementada anteriormente son correctas o no y por qué (0.5 p.):

A) `Equipo=leer();`      C) `leer(Equipo[100]);`      E) `leer();`  
B) `leer(Equipo);`      D) `Equipo[100]=leer();`      F) `Equipo[10]=leer();`

- Escriba una función para buscar los datos de una persona en el array. La función buscará por el nombre de la persona y devolverá todos los datos y un valor entero (1 si ha encontrado a la persona y 0 si no lo ha encontrado) (1 p.).  
`int buscar(struct persona P[], int tamanno, char nombre[], struct persona *pers);`
- Escriba una función en la que se utilice la función anterior. Declare las variables que sean necesarias, pida el nombre de la persona y llame adecuadamente a la función (0.5 p.).

4. Un cuadrado mágico es una matriz cuadrada con un número impar de filas y columnas, cuyas filas y columnas (e incluso sus diagonales) suman el mismo valor. Por ejemplo la matriz siguiente es un cuadrado mágico de 3 x 3:

6	1	8
7	5	3
2	9	4

⇒ Los números en cada fila, cada columna y cada diagonal suman 15.

Hacer un programa que compruebe si una matriz es un cuadrado mágico o no. El tamaño máximo de la matriz será de 20 x 20. En primer lugar el programa leerá el tamaño N de la matriz hasta que éste sea correcto (impar y menor que el tamaño máximo), y pedirá los datos de una matriz N x N. Una vez leídos y almacenados los datos se comprobará si la matriz es un cuadrado mágico o no y se mostrará el mensaje correspondiente (2.5 p.).

5. Conteste razonadamente a las siguientes preguntas (1.5 p.):

- ¿Es legal sumar un tipo `int` a un tipo `char`? ¿Cómo se denomina a este tipo de operación?
- ¿El obligatorio poner la sentencia `default` en un `switch`? ¿Y `break`?
- Dadas las declaraciones de la izquierda, responda a las cuestiones de la derecha:

```
int dato, *puntero;
puntero=&dato;
*puntero=5;
```

- ¿Cuál es el valor de `puntero`?
- ¿Cuál es el valor de `dato`?
- ¿Cuál es el valor de `*puntero`?

- Si se pierde el carácter '`\0`' del final de una cadena de caracteres, ¿qué hace la función `strlen()`?
- ¿Cuántos caracteres pone en memoria la siguiente sentencia? `char cadena[]="abcdefghijklm"`