E.T.S. en Ingeniería Informática
Grado en Informática
Dpto. Ingeniería Informática y Sistemas

Problemas del tema: Funciones (más....)

1. ¿Cuál es la importancia de la función *main*?. ¿Por qué todos los programas deben tener esta función?.

2. Si *funcion1* es una función definida en el programa como

```
void funcion1(int a, int b);
```

   ¿puede aparecer la siguiente sentencia en el programa principal?

```
A = 2 * funcion1(0, 4);
```

3. Cuál es la salida del siguiente programa?

```
#include <stdio.h>

void MiFuncion(int z);       /* Prototipo de la función */

int main(void) {
    int z = 2;

          /* Se hacen 2 llamadas a la función 'MiFuncion' */
    MiFuncion(z);
    MiFuncion(z);
}

void MiFuncion(int z) {       /* Definición de la función */
    printf("%d\n", z);
    z++;
}       /* Fin de la función 'MiFuncion' */
```

4. Cuál es la salida del siguiente programa?

```
#include <stdio.h>

int a, b, c;

/* Esta función se encarga de asignar valores a todas las variables */
void AsignacionValores(void) {
    int b;

    a = 1;
    b = 2;
    c = 3;
}           /* Fin de la función 'AsignacionValores' */

int main(void) {
        /* Se asignan valores iniciales a las variables antes de llamar a
           la función de 'AsignacionValores' */
    a = 101;
    b = 102;
    c = 103;
    AsignacionValores();
    printf("a =%d\t b =%d\t c =%d\n", a, b, c);
}
```

5. Cuál es la salida del siguiente programa?

```
#include <stdio.h>

int Multiplicar(int d, int b);       /* Prototipo de la función */

int d = 5;       /* Variable global */

int main(void) {
    int a, b, c;         /* Variables locales */
    int e = 4;

    a = 1;
    b = 2;
    c = Multiplicar(a, b);
    printf("a =%d\t b =%d\t c =%d\t d =%d\t e =%d\n", a, b, c, d, e);
}
```

```
/* Función que devuelve el resultado de multiplicar dos valores */
int Multiplicar(int d, int b) {
    int a;      /* Variable local */

    a = 2;
    b = 3;
    return (a * b * d);
}           /* Fin de la función 'Multiplicar' */
```

6. Cuál es la salida del siguiente programa si los valores introducidos como entrada son 2 10 3?

```
#include <stdio.h>

int f(int x, int y, int z);           /* Prototipos de las funciones */
int g(int arg);
int h(int arg1, int arg2);

int main(void) {
    int a, b, c;

    printf("Introduce 3 números separados por espacios en blanco: ");
    scanf("%d %d %d", &a, &b, &c);
    printf("Resultado de la función f =%d\n", f(a, b, c));
}

int f(int x, int y, int z) {          /* Definición de la función 'f' */
    int x1;

    x1 = g(x);
    return h(y, z) * x1;
}

int g(int arg) {                      /* Definición de la función 'g' */
    return arg * arg;
}

int h(int arg1, int arg2) {           /* Definición de la función 'h' */
    return arg1 / arg2;
}
```

7. Dadas las declaraciones siguientes:

```
double x, y, z;
int m, n;

void Hacer(int A, int B, int x);
```

¿Cuáles de las siguientes llamadas a la función son incorrectas y por qué?

a) `Hacer(x, y, z);`

b) `Hacer(x, y, M, 10);`

c) `Hacer(A, B, x);`

d) `Hacer(10.2, 1, m);`

8. ¿Cuál es la salida del siguiente programa?. Explica por qué la función *Swap* hace lo que hace

```
#include <stdio.h>

void Swap(int y, int x);        /* Prototipo de la función */

int main(void) {
    int x = 1;
    int y = 2;

    Swap(x, y);
    printf("x =%d\t y =%d\n", x, y);
}

/* Función que se encarga de intercambiar dos valores */
void Swap(int y, int x) {
    int temp;

    temp = x;
    x = y;
    y = x;
}           /* Fin de la función 'Swap' */
```

¿Cómo se puede cambiar el programa para que los cambios de realizados en la función *Swap* se mantengan?.

9. ¿Cuál es la salida del siguiente programa?

```
#include <stdio.h>

int a, b, c;      /* Variables globales */

void Calculo (int *x, int *y, int z); /* Prototipo de la función */

void Calculo (int *x, int *y, int z) {
    int a;        /* Variable local */

    a = 10;
    b = 5:
    *x = *x + 1;
    *y = *y + 2;
    z = z + 3;
    printf("a =%d\t b =%d\t x =%d\t y =%d\t z =%d\n", a, b, *x, *y, z);
}           /* Fin de la función Calculo */

int main(void) {
    a = 1;
    b = 2;
    c = 3;
    Calculo(&a, &b, c);
    printf("a =%d\t b =%d\t c =%d\n", a, b, c);
    Calculo(&c, &a, c);
    printf("a =%d\t b =%d\t c =%d\n", a, b, c);
}
```

10. Escribir una función que reciba un número como parámetro y devuelva su cuadrado.

11. Escribir una función que, dados dos números enteros, devuelva su suma.

12. Escribir una función que convierta una temperatura en grados Centígrados en su temperatura equivalente en grados Fahrenheit.

$$(9/5) * c + 32$$

donde c: temperatura en grados Centígrados
y f: temperatura en grados Fahrenheit.

13. Escribir una función que calcule el área de un cilindro dado ($Area = 2 * \pi * radio * altura$).

14. Escribir una función que calcule la longitud de la hipotenusa de un triángulo rectángulo dados sus dos catetos. ($hipotenusa^2 = lado1^2 + lado2^2$).

15. Escribir una función que reciba dos números como parámetros y devuelva el máximo de los dos.

16. Escribir una función que tenga un argumento de tipo entero y que devuelva la letra P si el número es positivo y la letra N si el número es cero o negativo.

17. Escribir una función que reciba como parámetro un valor de tipo entero y que devuelva un 1 si el número es múltiplo de 5 y un 0 si no lo es.

18. Escribir una función que devuelva un 1 si se le pasa como parámetro un cero o un número x comprendido en el rango: $50 <= x < 100$; y que devuelva un 0 en caso contrario.

19. Escribir una función que reciba un año como parámetro y devuelva un 1 si el año es bisiesto y un 0 si no lo es. Un año es bisiesto si es múltiplo de 4, excepto los múltiplo de 100, que no son bisiestos, salvo que a su vez también sean múltiplos de 400.

20. Escribir una función que reciba un número positivo y devuelva el factorial de ese número ($num! = num * (num - 1) * (num - 2) * \ldots * 1$).

21. Crear una función que pase una cadena de mayúsculas a minúsculas o viceversa dependiendo de un valor 0/1 (*Mayus*) que se le pase como parámetro.
Nota: Existen 2 funciones en la librería *ctype.h* que permite pasar un caracter de mayúscula a minúscula y viceversa:
    - *char toupper(char c);* : Permite pasar un caracter a mayúscula.
    - *char tolower(char c);* : Permite pasar un caracter a minúscula.

22. Escribir una función que reciba un número positivo, n, y muestre por pantalla todos los números entre 0 y n.

23. Escribir una función que reciba un número entre 1 y 10 y muestre por pantalla la tabla de multiplicar de ese número.

24. Escribir una función que devuelva la suma de los enteros entre 1 y 10: $1+2+3+\ldots+9+10$.

25. Escribir una función que imprima por pantalla todas las potencias de 2 entre 1 y 10000.

26. Escribir una función que lea números negativos de teclado y devuelva el número de valores leídos. La lista terminará cuando se lea un 0 o un número positivo.

27. Escribir una función que dado un array de N valores enteros, calcule la suma de todos los valores.

28. Escribir una función que dado un array de N valores enteros, muestre todos los valores por pantalla.

29. Escribir una función que devuelva el valor máximo de un array de N valores que se le pasa como parámetro.

30. Escribir una función que reciba un vector de 10 números, calcule el doble de cada número y devuelva el resultado obtenido en un nuevo vector.

31. Escribir una función que reciba como parámetros dos vectores de 20 elementos cada uno y devuelva el vector suma, cuyo valor en la posición i se calcula como la suma de los valores que se encuentran en la posición i de los dos vectores.

32. Escribir una función que reciba un vector de N elementos y sustituya en ese vector todos los valores menores que 5 por ceros.

33. Escribir una función que reciba como parámetro un array de N números flotantes ordenados de menor a mayor, un entero que indique cuántos de los elementos del array están ocupados y otro número flotante. La función debe buscar el número en el array. Si el número está debe mostrar un mensaje por pantalla indicando la posición que ocupa en el array. Si el número no está debe insertarlo de forma ordenada en el array. La función debe devolver el número de elementos que hay en el array.

34. Escribir una función que reciba 2 array de N enteros ordenados de menor a mayor, y mezcle los dos arrays en uno nuevo también ordenado.

    Ejemplo:
    array1: 1 3 5 8 11 21
    array2: 2 7 9 10 22 26
    resultado: 1 2 3 5 7 8 9 10 11 21 22 26

35. Escribir una función que reciba una matriz de NxN elementos y la muestre por pantalla con la forma de matriz (filas y columnas).

36. Escribir una función que pida datos por teclado y los almacene en una matriz de tamaño NxM. La matriz no puede ser una variable global.

37. Escribir una función que reciba una matriz de NxN elementos y calcule la suma de su diagonal principal.

38. Escribir una función que reciba una matriz de de tamaño NxM y devuelva el número de veces que aparece el número 0.

39. Dada una matriz cuadrada: *int matriz[N][N];* escribir funciones para determinar si:

    a) La matriz es simétrica: $aij = aji$
    b) La matriz es antisimétrica: $aij = -aji$
    c) La matriz es triangular inferior: Todos los elementos por encima de la diagonal principal son cero.

40. Escribir funciones para calcular la traspuesta de una matriz, la suma y el producto de dos matrices.

41. Dada una matriz de números enteros comprendidos entre 1 y 100 escribir una función que reciba la matriz como parámetro, detecte todos los números que están repetidos y los reemplace por cero, indicando cuántos hay sin repetir.

42. Escribir una función que reciba como parámetros una matriz de tamaño NxM flotantes y un entero que represente el índice de una fila. La función debe devolver el máximo de los elementos de esa fila.

43. Escribir una función que reciba una matriz de tamaño NxM y calcule la suma de los elementos por columnas, devolviendo los resultados en un vector.

44. Escribir una función que reciba una matriz cuadrada como parámetro y calcule si esa matriz es mágica: la suma de los elementos cada una de sus filas, de cada una de sus columnas y de sus diagonales, tienen el mismo valor. Debe devolver un 1 si lo es y un 0 si no lo es.

45. Escribir una función que reciba como datos la longitud y anchura de un rectángulo y devuelva su área y su perímetro.

46. Escribir una función para convertir las coordenadas polares (r, σ) de un punto P a las coordenadas rectangulares (x, y), donde: $x = r * cos\sigma$, $y = r * sen\sigma$.

47. Escribir una función que dada una cantidad positiva de segundos, devuelva su equivalente en horas, minutos y segundos.

48. Escribir una función que reciba dos números enteros, x e y, y devuelva su producto, cociente y resto.

49. Escribir una función que reciba como parámetro un número entero y devuelva su cuadrado y su cubo.

50. Escribir una función que reciba un vector de N elementos y devuelva el número de valores igual a 0 y el número de valores positivos.

51. Escribir una función que reciba como parámetro un array de N números flotantes, devuelva el máximo y el mínimo de los valores del array.

52. Escribir un programa que encuentre el valor mayor, el valor menor, la suma y la media de los datos de entrada. Los dos últimos valores (la suma y la media) se obtendrán mediante una función. Los datos se encontrarán almacenados en un vector.

53. Escribir una función que reciba como parámetro un array de N elementos, donde cada elemento es una estructura con información sobre alumnos (nombre, apellido y nota). La función debe rellenar 2 arrays, uno que contenga los alumnos aprobados (la nota sea $>= 5$) y otro con los alumnos suspendidos (la nota sea < 5).

54. Considerando la estructura para representar libros y revistas de una biblioteca, vista en la hoja de problemas del tema *Variables y Operadores*, escribir una función que reciba como parámetros un array de libros que contiene una biblioteca, el número de libros existentes y el nombre de un autor y devuelva el primer libro encontrado de ese autor y el número total de libros de ese autor.

55. Se tiene un vector de estructuras con los datos de los empleados de una empresa (nombre, apellidos, año de contrato en la empresa y sueldo). El vector no está completamente lleno, se reconoce los elementos del vector que no están rellenos porque tienen en el campo sueldo un valor -1. Escribir una función que reciba como parámetro el vector de empleados y devuelva la suma total de los sueldos.

56. Escribir un programa que reciba 5 parámetros y los muestre en sentido inverso (desde el último parámetro al primero). Si el número de parámetros pasados no es correcto debe mostrar un mensaje que lo comunique.

57. Escribir un programa que reciba dos números como parámetros y muestre por pantalla la suma y la resta de esos dos valores.

58. Escribir un programa que reciba 3 parámetros. El primero debe ser un operador aritmético ($+$, $-$, $*$, $/$) y los dos parámetros siguientes serán números enteros. En el caso de que el número de parámetros no sea correcto o que el primer parámetro no sea uno de los operadores válidos, el programa debe mostrar por pantalla un mensaje de error, si no, mostrará el resultado de realizar la operación.

    Ejemplo: Si el programa se llama *calculadora* y la ejecución se realiza con el comando: *./calculadora * 2 3*
    El resultado que debe mostrarse por pantalla es: *El resultado de operar 2 * 3 = 6*