

**Ampliación de Física del Estado Sólido**  
Relación 1: Introducción al lenguaje C

**Juan J. Meléndez**

## Problema 1.1

Si, en un programa en C, aparecen las líneas

```
int i = 10, j = 4, k, n;
double a, b, c;
```

indicar el valor de las siguientes expresiones:

- a)  $a = i + 3.5*j$ ;                      c)  $n = a - i/j$ ;                      e)  $c = a - i/j$ ;  
 b)  $k = (a - i)/j$ ;                      d)  $b = (a - i)/j$ ;

La salida que pide el enunciado es:

- a) 24.0;                      b) 3;                      c) 22;                      d) 3.5;                      e) 21.5

Dependiendo de las opciones de optimización del compilador, el resultado puede variar en algunos casos. La salida que se da corresponde a la opción estándar `gcc -O2`.

En los siguientes problemas, la solución que se da corresponde a un código que hace lo especificado en el enunciado y que funciona. No es posible dar una solución única y, por tanto, cualquier otra solución que funcione es válida.

## Problema 1.2

Escribir un código C que utilice los comandos `printf` y `scanf` para leer una variable entera `a` y otra entera `k`, precedidas de mensajes por pantalla adecuados. A continuación, escribir los valores de `a` y `k` multiplicadas por 3.14, precedidos de una frase explicatoria de lo que se muestra.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, const char * argv[]) {
    int a, k;
    double a2, k2;
    double pi = 3.14;

    printf("Introducir dos valores enteros a y k:\n");
    scanf("%d %d", &a, &k);

    a2 = pi*a;
    k2 = pi*k;

    printf("El valor 3.14*a es %lf \n", a2);
    printf("El valor 3.14*k es %lf \n", k2);
    return 0;
}
```

### Problema 1.3

Escribir un código C que calcule la posición en un cierto instante  $t$  de una partícula lanzada con una velocidad inicial  $\vec{v}_0$  que forma un ángulo  $\theta$  con la horizontal. El programa debe solicitar los parámetros de entrada  $v_0$ ,  $\theta$  y  $t$ , que deben ser introducidos por el terminal. La salida del programa, en un archivo, deben ser los valores de  $x$ ,  $y$  y  $t$ , con el formato adecuado.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main(int argc, const char * argv[]) {
    double t, v_0, theta;
    double x, y;
    double t_0;
    double g = 9.8;

    printf("Introducir el modulo de la velocidad inicial en m/s: \n");
    scanf("%lf", &v_0);
    printf("Introducir el angulo de inclinacion en grados:\n");
    scanf("%lf", &theta);
    printf("Introducir el instante de tiempo:\n");
    scanf("%lf", &t);

    theta *= 2.*M_PI/360;
    t_0 = 2*v_0*sin(theta)/g;

    if (t < t_0){
        x = v_0*cos(theta)*t;
        y = v_0*sin(theta)*t - 0.5*g*t*t;
    } else {
        x = v_0*v_0*sin(2.*theta)/g;
        y = 0.;
    }

    printf("En el instante %lf, la partícula se encuentra en la posición
           (x,y) = (%lf, %lf).", t, x, y);
    return 0;
}
```

### Problema 1.4

Escribir un programa C que:

*a)* calcule el factorial de un número natural (incluido el cero); el programa debe lanzar un mensaje de error si el factorial no se puede calcular.

*b)* resuelva la ecuación

$$\cos x + x = 0$$

por el método de la bisección. El programa debe mostrar el resultado en pantalla.

- c) busque los números primos comprendidos entre 1 y  $N$ , donde  $N$  debe ser suministrado por pantalla. El programa debe mostrar la lista de números primos por pantalla.

```
a) #include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main(int argc, const char * argv[]) {
    int fact;
    int j;
    double n;
    double aux;

    printf("Introducir un numero natural\n");
    scanf("%lf", &n);

    if (modf(n, &aux) != 0 || n < 0) {
        printf("El numero introducido no es correcto.
                No se puede calcular el factorial de un numero
                no natural. Saliendo...\n");
        exit(1);
    } else if (n == 0) fact = 1;
    else {
        fact = 1;
        for(j = 1; j <= n; j++) fact *= j;
    }
    printf("El factorial pedido es %d\n", fact);
    return 0;
}
```

```
b) #include <stdio.h>
#include <stdlib.h>
#include <math.h>

double function (double x){
    double result = cos(x) + x;
    return result;
}

int main(int argc, const char * argv[]){
    double b = 0, a = -1.57;
    double r = (a + b) / 2.;

    while (fabs(function(r)) > 0.000001){
        if (function(a) == 0){
            r = a;
        }
    }
}
```

```
        } else if (function(b) == 0){
            r = b;
        } else if (function(a)*function(r)>0){
            a = r;
        } else {b = r;}
        r = (a + b) / 2.;
    }

    printf("La raiz de la ecuacion es %f\n", r);
    return 0;
}

c) #include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main(int argc, const char * argv[]){
    double N, aux;
    int i, j;

    printf("Introducir un numero natural positivo\n");
    scanf("%lf", &N);

    if (modf(N, &aux) != 0 || N <= 0) {
        printf("El numero introducido no es
            natural positivo. Saliendo...\n");
        exit(1);
    } else {
        printf("Los numeros primos comprendidos
            entre 1 y %d son:\n", (int)N);
        for (i = 2; i <= N; i++){
            j = 2;
            while (i%j != 0) j++;
            if (j >= i && i < N) printf("%d, ", i);
            if (i == N) printf("%d.", i);
        }
        printf("\n");
    }
    return 0;
}
```

## Problema 1.5

Escribir un código C que pregunte por pantalla si se desea pasar de escala Celsius a Fahrenheit o viceversa, realice el cambio de escala solicitado y, a continuación, muestre el resultado en pantalla.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, const char * argv[]) {
    float celsius, fahrenheit;
    int cambio;

    printf("Si desea pasar de escala Celsius a Fahrenheit pulse 1.
           Si se desea pasar de escala Fahrenheit a Celsius pulse 2.
           Si no desea realizar ningun cambio de escala
           pulse cualquier otro numero. \n");
    scanf("%d", &cambio);

    switch (cambio){
        case 1:

            printf("Introduzca la temperatura en grados Celsius.\n");
            scanf("%f", &celsius);

            fahrenheit = 9.*celsius/5. + 32.;
            printf("La temperatura en grados Fahrenheit es
                   %f F\n", fahrenheit);

            break;

        case 2:

            printf("Introduzca la temperatura en grados Fahrenheit.\n");
            scanf("%f", &fahrenheit);

            celsius = (fahrenheit - 32.)*5./9.;
            printf("La temperatura en grados Celsius es
                   %f C\n", celsius);

            break;

        default:

            printf("Saliendo ... \n");
            exit(1);
    }
    return 0;
}
```

## Problema 1.6

Escribir un código C que calcule el valor máximo, mínimo y medio de una lista de  $N$  números reales, donde  $N$  es un parámetro del programa, suministrado por un archivo. Los valores máximo, mínimo y medio deben mostrarse en otro archivo, junto con una frase que explique qué es cada uno.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, const char * argv[]) {
    int N, i;
    float *datos, max, min, media;

    FILE *ficherodatos, *ficheroresultados;

    ficherodatos = fopen("lista.txt", "r");
    ficheroresultados = fopen("resultados.txt", "w");

    fscanf(ficherodatos, " N = %d\n", &N);

    datos = (float *)calloc(N, sizeof(float));

    for (i = 0; i < N; i++){
        fscanf(ficherodatos, "%f\n", &datos[i]);
    }

    fclose(ficherodatos);

    max = datos[0];
    min = datos[0];
    media = datos[0];

    for (i = 1; i < N; i++){

        if (datos[i] < min){min = datos[i];}
        if (datos[i] > max){max = datos[i];}
        media += datos[i];
    }

    media = media/N;

    fprintf(ficheroresultados, "El valor minimo es %f, el maximo es %f
    y el valor medio es %f\n", min, max, media);

    return 0;
}
```

## Problema 1.7

Indicar, razonadamente, si las siguientes expresiones son correctas o no:

a) <code>int a;</code> <code>double b, *pd;</code> <code>pd = &amp;a;</code>	d) <code>double b, *pd;</code> <code>*pd = 3.14;</code> <code>b = *pd;</code>	<code>*pd;</code> <code>pd = &amp;b;</code> <code>a = *pd;</code>
b) <code>double b, c, *pd;</code> <code>b = 3.14;</code> <code>c = &amp;(*pd);</code>	e) <code>double b, c, *pd;</code> <code>b = 3.14;</code> <code>pd = &amp;c;</code> <code>*pd = &amp;b;</code>	g) <code>double b, c, *pd;</code> <code>b = 3.14;</code> <code>pd = &amp;c;</code> <code>*pd = b;</code>
c) <code>double b, *pd;</code> <code>pd = &amp;b;</code> <code>*pd = 3.14;</code>	f) <code>int a;</code> <code>double b = 3.14,</code>	h) <code>int a;</code> <code>a = &amp;2;</code>

- a) Incorrecta: la variable `pd` es un puntero a `double` y, por tanto, no se puede igualar a la dirección de `a`, que es un puntero a entero.
- b) Incorrecta, ya que iguala `c`, una variable tipo `double`, a la dirección de `*pd`, que es un puntero.
- c) Correcta.
- d) Correcta.
- e) Incorrecta, porque iguala la indirección de `pd`, una variable `double`, a la dirección de `b`, un puntero a `double`.
- f) Incorrecta, ya que iguala la variable entera `a` a la indirección de `pd`, que es de tipo `double`.
- g) Correcta.
- h) Incorrecta, ya que `a` es una variable de tipo entero y la dirección de `2` un puntero.

## Problema 1.8

Escribir un código C que realice el ajuste por mínimos cuadrados de la siguiente serie de puntos experimentales:

$x$	5	7	10	12	16	20	23	27	19	14	9	6
$y$	9	11	15	16	20	24	27	29	22	20	14	9

La pendiente y ordenada en origen de la recta de mejor ajuste están dados por

$$m = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}$$

$$a = \bar{y} - m\bar{x},$$

respectivamente, donde  $\bar{x}$  y  $\bar{y}$  son los valores medios de las variables independiente y dependiente. El coeficiente de correlación está dado por:

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

El programa debe mostrar el resultado del análisis en un archivo, y debe resolverse utilizando punteros.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

float media(int *datos){
    float result = 0;

    for (int i = 0; i < 12; i++) result += datos[i];

    result /= 12.;
    return result;
}

int main(int argc, const char * argv[]){
    int x[12] = {5, 7, 10, 12, 16, 20, 23, 27, 19, 14, 9, 6};
    int y[12] = {9, 11, 15, 16, 20, 24, 27, 29, 22, 20, 14, 9};
    float m, a, r, xmedia, ymedia;
    float sumaxx = 0, sumaxy = 0, sumayy = 0;

    FILE *resultados;
    resultados = fopen("regresion.txt", "w");

    xmedia = media (x);
    ymedia = media (y);

    for (int i = 0; i < 12; i++){
        sumaxx += (x[i] - xmedia) * (x[i] - xmedia);
        sumaxy += (x[i] - xmedia) * (y[i] - ymedia);
        sumayy += (y[i] - ymedia) * (y[i] - ymedia);
    }

    m = sumaxy / sumaxx;
    a = ymedia - m * xmedia;
    r = sumaxy / sqrt(sumaxx * sumayy);

    fprintf(resultados, "La pendiente es m = %f, la ordenada en el
        origen es a = %f y el coeficiente de correlacion es
        r = %f\n", m, a, r);

    return 0;
}
```