

FACULTAD DE INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VARIABLES SIMPLES EN LENGUAJE C

/* VS01: Escribir un programa en lenguaje C que:

Lea los números enteros a, b, c y d, e imprima luego dichos números y; a) Si son positivos, un mensaje que indique, para cada uno de ellos, si es par o impar; b) Un mensaje de error en caso contrario */

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

```
void cls(void);
```

```
main()
```

```
{
```

```
    int a, b, c, d;
```

```
    cls();
```

```
    printf("\n\t\t\t\tFacultad de Ingeniería - COMPUTACIÓN - Ejercicio VS01\n\n");
```

```
    printf("\n\t\tDigite cuatro números enteros positivos: ");
```

```
    scanf("%d %d %d %d", &a, &b, &c, &d);
```

```
    if (a > 0 && b > 0 && c > 0 && d > 0)
```

```
    { if (a%2 == 0)
```

```
        { printf("\n\t\tEl primer número %d es par", a); }
```

```
        else
```

```
        { printf("\n\t\tEl primer número %d es impar", a); }
```

```
        if (b%2 == 0)
```

```
        { printf("\n\t\tEl segundo número %d es par", b); }
```

```
        else
```

```
        { printf("\n\t\tEl segundo número %d es impar", b); }
```

```
        if (c%2 == 0)
```

```
        { printf("\n\t\tEl tercer número %d es par", c); }
```

```
        else
```

```
        { printf("\n\t\tEl tercer número %d es impar", c); }
```

```
        if (d%2 == 0)
```

```
        { printf("\n\t\tEl cuarto número %d es par\n\n", d); }
```

```
        else
```

```
        { printf("\n\t\tEl cuarto número %d es impar\n\n", d); } }
```

```
    else
```

```
    { printf("\n\t\tLos números dados %d, %d, %d, %d no son todos positivos\n\n", a, b, c, d); }
```

```
}
```

```
void cls(void)
```

```
{
```

```
    printf("\033[2J"); /* Limpia la pantalla */
```

```
    printf("\033[0;0f"); /* Mueve el cursor en la esquina superior izquierda */
```

```
}
```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VARIABLES SIMPLES EN LENGUAJE C

/* VS02: Escribir un programa en lenguaje C que:

Lea los números enteros a, b, c y d e imprima luego dichos números y;

a) Si son positivos, un mensaje que indique, para cada uno de ellos, si es o no múltiplo de 5; b) Un mensaje de error en caso contrario. */

```
#include <stdio.h>
void cls(void);
main()
{
    int a, b, c, d;
    cls();
    printf("\n\t\t\tFacultad de Ingeniería - COMPUTACIÓN - Ejercicio VS02\n\n");
    printf("\n\t\tDigite cuatro números enteros positivos: ");
    scanf("%d %d %d %d", &a, &b, &c, &d);
    if (a > 0 && b > 0 && c > 0 && d > 0)
    { if (a%5 == 0)
        { printf("\n\t\tEl primer número %d es múltiplo de 5", a); }
        else
        { printf("\n\t\tEl primer número %d no es múltiplo de 5", a); }
        if (b%5 == 0)
        { printf("\n\t\tEl segundo número %d es múltiplo de 5", b); }
        else
        { printf("\n\t\tEl segundo número %d no es múltiplo de 5", b); }
        if (c%5 == 0)
        { printf("\n\t\tEl tercer número %d es múltiplo de 5", c); }
        else
        { printf("\n\t\tEl tercer número %d no es múltiplo de 5", c); }
        if (d%5 == 0)
        { printf("\n\t\tEl cuarto número %d es múltiplo de 5\n", d); }
        else
        { printf("\n\t\tEl cuarto número %d no es múltiplo de 5\n", d); } }
    else
    { printf("\n\t\tLos números dados %d, %d, %d, %d no son todos positivos\n", a, b, c, d); }
}
void cls(void)
{
    printf("\033[2J");           /* Limpia la pantalla */
    printf("\033[0;0f");         /* Mueve el cursor en la esquina superior izquierda */}
```

FACULTAD DE INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VARIABLES SIMPLES EN LENGUAJE C

/* VS03: Escriba un programa en lenguaje C que:

Lea los números enteros a, b, c y d e imprima luego dichos números y;

- a) Si son positivos, un mensaje que indique, si son o no términos de una progresión aritmética,
- b) Un mensaje de error en caso contrario. */

```
#include <stdio.h>
void cls(void);
main()
{
    int a, b, c, d;
    cls();
    printf("\n\t\t\tFacultad de Ingeniería - COMPUTACIÓN - Ejercicio VS03\n\n");
    printf("\n\t\tDigite cuatro números enteros positivos: ");
    scanf("%d %d %d %d", &a, &b, &c, &d);
    if ( a > 0 && b > 0 && c > 0 && d > 0 )
    { if ( b - a == c - b && c - b == d - c )
        { printf("\n\t\tLos números %d, %d, %d y %d son términos de una progresión aritmética", a, b, c, d); }
        else
        { printf("\n\t\tLos números %d, %d, %d y %d no son términos de una progresión aritmética", a, b, c, d); }
    }
    else
    { printf("\n\t\tLos números dados %d, %d, %d, %d no son todos positivos\n\n", a, b, c, d); }
}
void cls(void)
{
    printf("\033[2J");           /* Limpia la pantalla */
    printf("\033[0;0f");         /* Mueve el cursor en la esquina superior izquierda */
```

FACULTAD DE INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VARIABLES SIMPLES EN LENGUAJE C

/* VS04: Escribir un programa en lenguaje C que:

Lea los números enteros a, b, c y d e imprima luego dichos números y;

- a) Si son positivos, un mensaje que indique, si son o no términos de una progresión geométrica,
- b) Un mensaje de error en caso contrario. */

```
#include <stdio.h>
void cls(void);
main()
{
    int a, b, c, d;
    cls();
    printf("\n\t\t\t\tFacultad de Ingeniería - COMPUTACIÓN - Ejercicio VS04\n\n");
    printf("\n\t\tDigite cuatro números enteros positivos: ");
    scanf("%d %d %d %d", &a, &b, &c, &d);
    if ( a > 0 && b > 0 && c > 0 && d > 0 )
    { if ( b/a == c/b && c/b == d/c )
        { printf("\n\t\tLos números %d, %d, %d y %d son términos de una progresión geométrica", a, b, c, d); }
        else
        { printf("\n\t\tLos números %d, %d, %d y %d no son términos de una progresión geométrica", a, b, c,
d); } }
    else
    { printf("\n\t\tLos números dados %d, %d, %d, %d no son todos positivos\n\n", a, b, c, d); }
}
void cls(void)
{
    printf("\033[2J");           /* Limpia la pantalla */
    printf("\033[0;0f");         /* Mueve el cursor en la esquina superior izquierda */
}
```

FACULTAD DE INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VARIABLES SIMPLES EN LENGUAJE C

/* VS05: Escribir un programa en lenguaje C que:

Lea los números enteros a, b, c y d e imprima luego dichos números y;

- a) Si son positivos, un mensaje que indique, si están o no ordenados en secuencia ascendente,
- b) Un mensaje de error en caso contrario. */

```
#include <stdio.h>
void cls(void);
main()
{
    int a, b, c, d;
    cls();
    printf("\n\t\t\tFacultad de Ingeniería - COMPUTACIÓN - Ejercicio VS05\n\n");
    printf("\n\t\tDigite cuatro números enteros positivos: ");
    scanf("%d %d %d %d", &a, &b, &c, &d);
    if ( a > 0 && b > 0 && c > 0 && d > 0 )
    { if ( a < b && b < c && c < d )
        { printf("\n\t\tLos números %d, %d, %d y %d están en secuencia ascendente\n\n", a, b, c, d); }
        else
        { printf("\n\t\tLos números %d, %d, %d y %d no están en secuencia ascendente\n\n", a, b, c, d); } }
    else
    { printf("\n\t\tLos números dados %d, %d, %d, %d no son todos positivos\n\n", a, b, c, d); }
}
void cls(void)
{
    printf("\033[2J");           /* Limpia la pantalla */
    printf("\033[0;0f");         /* Mueve el cursor en la esquina superior izquierda */
}
```

FACULTAD DE INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VARIABLES SIMPLES EN LENGUAJE C

/* VS06: Escribir un programa en lenguaje C que:

Permita determinar el máximo común divisor de dos números enteros y positivos a y b, utilizando el algoritmo de Euclides. Imprima además el mínimo común múltiplo de los mismos. */

```
#include <stdio.h>
void cls(void);
main()
{
    int a, b, p, q, r, mcd, mcm;
    cls();
    printf("\n\t\t\tFacultad de Ingeniería - COMPUTACIÓN - Ejercicio VS06\n\n");
    printf("\n\t\t\t Digite dos números enteros y positivos ");
    scanf("%d %d", &a, &b);
    while ( a <= 0 || b <= 0 )
    { printf("\n\t\t Los números deben ser positivos, digite nuevamente, ");
        scanf("%d %d", &a, &b); }
    if (a < b)
    { p = b;
        q = a; }
    else
    { p = a;
        q = b; }
    r = p%q;
    while ( r != 0 )
    { p = q;
        q = r;
        r = p%q; }
    mcd = q;
    mcm = a*b/mcd;
    printf("\n\t\t\t mcd( %d, %d ) = %d", a, b, mcd);
    printf("\n\t\t\t mcm( %d, %d ) = %d \n", a, b, mcm);
}
void cls(void)
{
    printf("\033[2J");           /* Limpia la pantalla          */
    printf("\033[0;0f");         /* Mueve el cursor en la esquina superior izquierda */
}
```

FACULTAD DE INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VARIABLES SIMPLES EN LENGUAJE C

```
/* VS07: Escribir un programa en lenguaje C que:  
   Permita determinar si un número entero dado n es primo. */  
#include <stdio.h>  
#include <math.h>  
void cls(void);  
main()  
{  
    int a, ls, q, r;  
    cls();  
    printf("\n\t\t\tFacultad de Ingeniería - COMPUTACIÓN - Ejercicio VS07\n\n");  
    printf("\t\tDigite un número entero y positivo ");  
    scanf("%d", &a);  
    if (a > 0)  
    {  
        q = 2;  
        r = a%q;  
        ls = sqrt(a);  
        while (r != 0 && q < ls)  
        {  
            q = q + 1;  
            r = a%q; }  
        if (r != 0)  
            { printf("\n\n\tEl número %d es primo\n\n", a); }  
        else  
            { printf("\n\n\tEl número %d no es primo\n\n", a); } }  
    else  
        { printf("\n\n\tEl número %d debe ser positivo\n\n", a); }  
}  
void cls(void)  
{  
    printf("\033[2J");           /* Limpia la pantalla */  
    printf("\033[0;0f");          /* Mueve el cursor en la esquina superior izquierda */  
}
```

FACULTAD DE INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VARIABLES SIMPLES EN LENGUAJE C

```
/* VS08: Escribir un programa en lenguaje C que:  
Permita descomponer un número entero dado n en sus factores primos. */  
#include <stdio.h>  
void cls(void);  
main()  
{  
    int p, q, a, n;  
    int factor[10][2];  
    cls();  
    printf("\n\t\t\t\tFacultad de Ingeniería - COMPUTACIÓN - Ejercicio VS08\n\n");  
    printf("\t\tDigite un número entero y positivo ");  
    scanf("%d", &a);  
    while ( a <= 0 )  
    { printf("\tEl número debe ser positivo, digite nuevamente, ");  
        scanf("%d", &a); }  
    n = 0;  
    p = a;  
    q = 1;  
    while ( p > 1 )  
    { q = q + 1;  
        if ( p%q == 0 )  
        { n = n + 1;  
            factor[n][1] = q;  
            factor[n][2] = 1;  
            p = p/q;  
            while (p%q == 0)  
            { factor[n][2] = factor[n][2] + 1;  
                p = p/q; } } }  
    printf("\n\n\t\tFactor Exponente (De la descomposición en factores primos)\n");  
    for ( p = 1; p <= n; p++ )  
    { printf("\t\t\t%4i%10i\n", factor[p][1], factor[p][2]); }  
    printf("\n");  
}  
void cls(void)  
{  
    printf("\033[2J"); /* Limpia la pantalla */  
    printf("\033[0;0f"); /* Mueve el cursor en la esquina superior izquierda */  
}
```

FACULTAD DE INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VARIABLES SIMPLES EN LENGUAJE C

```
/* VS09: Escribir un programa en lenguaje C que:  
    Acepte como datos los coeficientes a, b y c, de una ecuación de segundo grado, e imprima sus raíces.  
*/  
#include <stdio.h> // Para poder llamar a scanf() y printf()  
#include <math.h> // Para poder utilizar funciones matemáticas  
void cls(void); // Función que permite limpiar la pantalla  
main()  
{  
    float a, b, c, d, x1, x2; // Declaración de variables como número de punto flotante  
    cls();  
    printf("Digite los coeficientes a, b y c de la ecuación de segundo grado: " );  
    scanf("%f %f %f", &a, &b, &c); // Al digitar, separar cada número con blancos  
    while ( a == 0 )  
    { printf("\nEl coeficiente a no puede ser cero, digite nuevamente la terna:");  
        scanf("%f %f %f", &a, &b, &c); }  
    d = b*b - 4*a*c;  
    if ( d < 0 )  
    {  
        x1 = -b/2/a;  
        x2 = sqrt((-d))/2/a;  
        printf("Las raíces son : %8.3f ± %8.3fi\n", x1, x2);  
    }  
    else  
    {  
        d = sqrt(d);  
        x1 = (-b - d)/2/a;  
        x2 = (-b + d)/2/a;  
        printf("Las raíces son : %8.3f y %8.3f\n", x1, x2);  
    }  
}  
void cls(void) // void cls(void) porque no devuelve valor y no tiene argumentos  
{  
    printf("\033[2J"); /* Limpia la pantalla */  
    printf("\033[0;0f"); /* Mueve el cursor en la esquina superior izquierda */  
}
```

FACULTAD DE INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VARIABLES SIMPLES EN LENGUAJE C

/* VS10: Escribir un programa en lenguaje C que:

Acepte como dato un número entero n e imprima dos números que representan, respectivamente, la suma de los números naturales pares e impares menores o iguales al número dado. */

```
#include <stdio.h> // Para poder llamar a scanf() y printf()
#include <math.h> // Para poder utilizar funciones matemáticas
void cls(void); // Función que permite limpiar la pantalla
main()
{
    int i, n, spar, simp; // Declaración de variables como número de punto flotante
    cls();
    printf("\n\t\tDigite un número entero y positivo: ");
    scanf("%d", &n); // Lectura del número natural.
    while ( n <= 0 )
    { printf("\n\t\tEl número debe ser positivo, digite nuevamente: ");
        scanf("%f", &n); }
    printf("\n");
    if ( n%2 == 0 )
    { spar = (n + 2)*n/4;
        simp = n*n/4;
        printf("\n\t\tLa suma de los pares menores o iguales a %d es %d \n", n, spar);
        printf("\n\t\tLa suma de los impares menores a %d es %d \n\n\n", n, simp); }
    else
    { spar = (n + 1)*(n - 1)/4;
        simp = (n + 1)*(n + 1)/4;
        printf("\n\t\tLa suma de los pares menores a %d es %d \n", n, spar);
        printf("\n\t\tLa suma de los impares menores o iguales a %d es %d \n\n\n", n, simp); }
}
void cls(void) // void cls(void) porque no devuelve valor y no tiene argumentos
{
    printf("\033[2J"); /* Limpia la pantalla */
    printf("\033[0;0f"); /* Mueve el cursor en la esquina superior izquierda */
}
```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VARIABLES SIMPLES EN LENGUAJE C

```
/* VS11: Escribir un programa en lenguaje C que:  
Acepte como dato tres números enteros y positivos, dd, mm, aa, verifique si constituyen una fecha válida en el  
formato día/mes/año e imprima la fecha del día siguiente. */  
#include <stdio.h>  
void cls(void);  
main()  
{  
    int dd, mm, aa, ds, ms, as, fv, ab;  
    int abis[12] = {31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};  
    int anbis[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};  
    cls();  
    printf ("\n\tDigite el día, mes y el año de una fecha: ");  
    scanf ("%02d %02d %02d", &dd, &mm, &aa);  
    // Verificación de la validez de la fecha leída  
    fv = 1;          // fv = 1 si la fecha es válida, 0 en caso contrario  
    ab = 0;          // ab = 1 si es año bisiesto, 0 en caso contrario  
    if ( aa%4 == 0 )  
    { ab = 1; }  
    if ( dd > 0 && mm > 0 && aa >= 0 )  
    { if ( mm < 13 )  
        { if ( ab == 1 )  
            { if ( dd > abis[mm - 1] )  
                { fv = 0; } }  
            else  
            { if ( dd > anbis[mm - 1] )  
                { fv = 0; } } }  
        else  
        { fv = 0; } }  
    else  
    { fv = 0; }  
    // Determinación de la fecha del día siguiente  
    if ( fv == 1 )  
    { ds = dd + 1;  
        if ( ab == 1 )  
        { if ( ds <= abis[mm - 1] )  
            { ms = mm;  
                as = aa; }  
            else  
            { ds = 1;  
                ms = mm + 1;  
                as = aa;  
                if ( ms > 12 )  
                { ms = 1;  
                    as = aa + 1; } } }  
        else  
        { if ( ds <= anbis[mm - 1] )  
            { ms = mm;  
                as = aa; }  
            else  
            { ds = 1;  
                ms = mm + 1;  
                as = aa;  
                if ( ms > 12 )  
                { ms = 1;  
                    as = aa + 1; } } }  
    printf ("\n\t\t%02d/%02d/%02d es una fecha válida", dd, mm, aa);  
    printf ("\t - \tLa fecha del día siguiente es %02d/%02d/%02d\n\n", ds, ms, as); }  
else  
{ printf ("\n\t\t%02d/%02d/%02d no es una fecha válida\n\n", dd, mm, aa); }  
}  
void cls(void)  
{  
    printf("\033[2J");           /* Limpia la pantalla */  
    printf("\033[0;0f");         /* Mueve el cursor en la esquina superior izquierda */  
}
```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VARIABLES SIMPLES EN LENGUAJE C

/* VS12: Escribir un programa en lenguaje C que:

Calcule las suma de los primeros cincuenta ($n = 1, 2, 3, \dots, 50$) términos de la serie: $\text{suma}(7\sin(3nx)\cos(nx))$, para $x = 2.5$. */

```
#include <stdio.h>
#include <math.h>
#define nmax 50
#define x 2.5
void cls(void);
main()
{
    int n;
    float suma;
    cls();
    suma = 0;
    for ( n = 1; n <= nmax; n++ )
    { suma += 7*sin(3*n*x)*cos(n*x); }
    printf ("\n\t\tLa suma de los %d primeros términos de la serie, para x = %4.2f, es %f\n\n", nmax, x, suma);
}
void cls(void)
{
    printf("\033[2J");           /* Limpia la pantalla          */
    printf("\033[0;0f");         /* Mueve el cursor en la esquina superior izquierda */
}
```

FACULTAD DE INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VARIABLES SIMPLES EN LENGUAJE C

/* VS13: Escribir un programa en lenguaje C que:

Calcule $\exp(x)$ a partir de la fórmula $\exp(x) = 1 + x + x^2/2! + x^3/3! + \dots$ tomando un número de términos, tal que el último sumando sea menor que $10E-5$. */

```
#include <stdio.h>
#include <math.h>
#define tol 0.00001
void cls(void);
main()
{
    int i, n;
    float factn, x, potx, suma, ultsum;
    cls();
    printf ("\n\t\t\tDigite el valor de x (potencia de e): ");
    scanf ("%f", &x);
    ultsum = x;
    suma = 1;
    n = 0;
    potx = 1;
    while ( ultsum > tol )
    { n++;
        potx *= x;
        factn = 1;
        for ( i = 1; i <= n; i++ )
        { factn *= i; }
        suma += potx/factn;
        ultsum = potx/factn; }
    printf("\n\n\t\tCon n = %d y el sumando no tomado = %f se tiene que exp(%f) = %f \n\n", n+1, ultsum, x,
    suma);
}
void cls(void)
{
    printf("\033[2J");           /* Limpia la pantalla */
    printf("\033[0;0f");         /* Mueve el cursor en la esquina superior izquierda */
}
```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VARIABLES SIMPLES EN LENGUAJE C

```
/* VS14: Escribir un programa en lenguaje C que:  
 Lea un conjunto de nmeros positivos e imprima luego el menor y mayor de ellos. */  
#include <stdio.h>  
void cls(void);  
main()  
{  
    int i, n;  
    float a, men, may;  
    cls();  
    printf ("\n\t\t\t\tDigite el número de datos a considerar: ");  
    scanf ("%d", &n);  
    printf ("\n\t\t\tDigite el primer dato a considerar : ");  
    scanf ("%f", &a);  
    men = a;  
    may = a;  
    for ( i = 1; i < n; i++ )  
    { printf ("\t\t\tDigite el siguiente dato a considerar: ");  
        scanf ("%f", &a);  
        if ( a < men ) men = a;  
        if ( a > may ) may = a; }  
    printf("\n\n\tDe los %d datos el menor es %f y el mayor es %f\n\n", n, men, may);  
}  
void cls(void)  
{  
    printf("\033[2J");           /* Limpia la pantalla */  
    printf("\033[0;0f");         /* Mueve el cursor en la esquina superior izquierda */  
}
```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VARIABLES SIMPLES EN LENGUAJE C

/* VS15: Escribir un programa en lenguaje C que:

Lea sucesivas ternas de números a, b y c, e imprima uno de los siguientes mensajes, “los números dados” y

según corresponda:

- a) “están en progresión aritmética”,
- b) “están en progresión geométrica”,
- c) “están en progresión armónica”,
- d) “no están en progresión aritmética ni geométrica ni armónica”.

La condición de fin es a = b = c = 0. */

```
#include <stdio.h>
void cls(void);
main()
{
    int lg;
    float a, b, c;
    cls();
    printf ("\n\t\t\tDigite tres números: ");
    scanf ("%f %f %f", &a, &b, &c);
    while ( a != 0 || b != 0 || c != 0 )
    { lg = 1;
        if ( 2*b == a + c )
        { lg = 0;
            printf ("\n\t\t\tLos números dados están en progresión aritmética"); }
        if ( b*b == a*c )
        { lg = 0;
            printf ("\n\t\t\tLos números dados están en progresión geométrica"); }
        if ( 2/b == 1/a + 1/c )
        { lg = 0;
            printf ("\n\t\t\tLos números dados están en progresión armónica"); }
        if ( lg == 1 )
        { printf ("\n\t\t\tLos números dados no están en progresión aritmética ni geométrica ni armónica"); }
        printf ("\n\t\t\tDigite tres números: ");
        scanf ("%f %f %f", &a, &b, &c);
    }
    printf("\n\n");
}
void cls(void)
{
    printf("\033[2J");      /* Limpia la pantalla          */
    printf("\033[0;0f");    /* Mueve el cursor en la esquina superior izquierda */
}
```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VECTORES EN LENGUAJE C

```
/* Vect01: Escribir un programa en lenguaje C que:  
 Resuelva el problema del cajero (Dar el vuelto con el menor número posible de billetes) */  
#include <stdio.h>  
void cls(void);  
main()  
{  
    int v, s, i, n, nbil[11];  
    int dbil[11] = {100000, 50000, 10000, 5000, 1000, 500, 100, 50, 10, 5, 1};  
    cls();  
    printf ("\n\t\t\t\t\tDigite el monto del vuelto a dar: ");  
    scanf ("%d", &v);  
    while ( v != 0 )  
    { s = v;  
        n = 0;  
        while ( s != 0 )  
        { nbil[n] = s/dbil[n];  
            s -= nbil[n]*dbil[n];  
            n++; }  
        printf("\n\t\t\t\t\tDebe entregar como vuelto de G %9d: ", v);  
        for (i = 0; i < n; i++ )  
        { if ( nbil[i] != 0 )  
            { printf ("\n\t\t\t\t\t\t\t%d billetes de %9d ", nbil[i], dbil[i]); } }  
        printf ("\n\t\t\t\t\tDigite el monto del vuelto a dar: ");  
        scanf ("%d", &v); }  
    printf("\n\n");}  
void cls(void)  
{  
    printf("\033[2J"); /* Limpia la pantalla */  
    printf("\033[0;0f"); /* Mueve el cursor en la esquina superior izquierda */  
}
```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VECTORES EN LENGUAJE C

```

/* Vect02: Escribir un programa en lenguaje C que:
Acepte como entrada un número natural n, las componentes de dos vectores u y v, de n
componentes, e imprima
el producto escalar de u por v. */
#define nmax 50
#include <stdio.h>
void cls(void);
main()
{
    int i, n;
    float s, u[nmax], v[nmax];
    cls();
    printf ("\n\t\t\t\tDigite el número de componentes de los vectores: ");
    scanf ("%d", &n);
    while ( n <= 0 )
    { printf ("\n\t\t\tEl número de componentes debe ser entero, positivo y menor que %d, digite
nuevamente: ", nmax);
        scanf ("%d", &n); }
    printf ("\n");
    for ( i = 0; i < n; i++ )
    { printf ("\t\t\tDigite la componente %d de cada vector: ", i);
        scanf ("%f %f", &u[i], &v[i]); }
    s = 0;
    for ( i = 0; i < n; i++ )
    { s += u[i]*v[i]; }
    printf ("\n\t\t\tEl producto escalar es %f\n\n\n", s);
}
void cls(void)
{
    printf ("\033[2J");      /* Limpia la pantalla */          */
    printf ("\033[0;0f");    /* Mueve el cursor en la esquina superior izquierda */
}

```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VECTORES EN LENGUAJE C

```
/* Vect03: Escribir un programa en lenguaje C que:  
Permita determinar la mayor y menor entre las n componentes numéricas de un vector e imprima las  
posiciones que ocupan  
los primeros de la secuencia que cumplen dichas condiciones. */  
#define nmax 50  
#include <stdio.h>  
void cls(void);  
main()  
{  
    int men, may, i, n;  
    float v[nmax];  
    cls();  
    printf ("\n\t\t\t\tDigite el número de componentes del vector: ");  
    scanf ("%d", &n);  
    while ( n <= 0 )  
    { printf ("\t\t\tEl número de componentes debe ser entero, positivo y menor que %d, digite  
nuevamente: ", nmax);  
        scanf ("%d", &n); }  
    printf ("\n");  
    for ( i = 0; i < n; i++ )  
    { printf ("\t\t\tDigite la componente %d del vector: ", i);  
        scanf ("%f", &v[i]); }  
    men = 0;  
    may = 0;  
    for ( i = 1; i < n; i++ )  
    { if ( v[i] < v[men] ) men = i;  
        if ( v[i] > v[may] ) may = i; }  
    printf ("\n\t\tLa menor de las componentes es v[%d] = %f", men, v[men]);  
    printf ("\n\t\tLa mayor de las componentes es v[%d] = %f\n", may, v[may]);  
}  
void cls(void)  
{  
    printf ("\033[2J");      /* Limpia la pantalla */  
    printf ("\033[0;0f");     /* Mueve el cursor en la esquina superior izquierda */  
}
```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VECTORES EN LENGUAJE C

```
/* Vect04: Escribir un programa en lenguaje C que:  
Permita determinar a) la suma, b) el producto y, c) la media aritmética, de las n componentes de  
un vector. */  
#define nmax 50  
#include <stdio.h>  
void cls(void);  
main()  
{  
    int i, n;  
    float suma, prod, m, v[nmax];  
    cls();  
    printf ("\n\t\t\t\tDigite el número de componentes del vector: ");  
    scanf ("%d", &n);  
    while ( n <= 0)  
    { printf ("\n\t\t\tEl número de componentes debe ser entero, positivo y menor que %d, digite  
nuevamente: ", nmax);  
        scanf ("%d", &n); }  
    printf ("\n");  
    for ( i = 0; i < n; i++ )  
    { printf ("\t\t\t\tDigite la componente %d del vector: ", i);  
        scanf ("%f", &v[i]); }  
    suma = v[0];  
    prod = v[0];  
    for ( i = 1; i < n; i++ )  
    { suma += v[i];  
        prod *= v[i]; }  
    m = suma/n;  
    printf ("\n\n\t\tLa suma de las componentes del vector es %f      ", suma);  
    printf ("\n\n\t\tEl producto de las componentes del vector es %f      ", prod);  
    printf ("\n\n\t\tLa media aritmética de las componentes del vector es %f\n\n", m);  
}  
void cls(void)  
{  
    printf ("\033[2J"); /* Limpia la pantalla */  
    printf ("\033[0;0f"); /* Mueve el cursor en la esquina superior izquierda */  
}
```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VECTORES EN LENGUAJE C

```
/* Vect05: Escribir un programa en lenguaje C que:  
Permita insertar, una componente dada, en la posición m en un vector de n componentes (n >= m).  
Luego de la  
inserción, el vector tendrá n + 1 componentes. */  
#define nmax 50  
#include <stdio.h>  
void cls(void);  
main()  
{  
    int i, n, m;  
    float a, v[nmax];  
    cls();  
    printf ("\n\t\t\t\t\tDigite el número de componentes del vector: ");  
    scanf ("%d", &n);  
    while ( n <= 0 )  
    { printf ("\n\t\t\tEl número de componentes debe ser entero, positivo y menor que %d, digite  
nuevamente: ", nmax - 1);  
        scanf ("%d", &n); }  
    printf ("\n");  
    for ( i = 0; i < n; i++ )  
    { printf ("\t\t\t\tDigite la componente %d del vector: ", i);  
        scanf ("%f", &v[i]); }  
    printf ("\n\t\tDigite la posición (0 <= m <= %d) y la componente a insertar en el vector: ", n);  
    scanf ("%d %f", &m, &a);  
    while ( m < 0 || m > n )  
    { printf ("\n\t\t\t\tPosición de componente a insertar en el vector incorrecta, digite nuevamente: ");  
        scanf ("%d %f", &m, &a); }  
    for ( i = n; i > m; i-- )  
    { v[i] = v[i-1]; }  
    v[m] = a;  
    printf("\n");  
    for ( i = 0; i <= n; i++ )  
    { printf ("\n\t\t\tLa componente %d del vector es %f: ", i, v[i]); }  
    printf("\n\n");}  
void cls(void)  
{  
    printf ("\033[2J");      /* Limpia la pantalla */  
    printf ("\033[0;0f");     /* Mueve el cursor en la esquina superior izquierda */  
}
```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VECTORES EN LENGUAJE C

```
/* Vect06: Escribir un programa en lenguaje C que:  
Permita eliminar la componente que ocupa la posición m en un vector de n componentes (n > m).  
Luego de la  
eliminación, el vector tendrá n – 1 componentes. */  
#define nmax 50  
#include <stdio.h>  
void cls(void);  
main()  
{  
    int i, n, m;  
    float v[nmax];  
    cls();  
    printf ("\n\t\t\t\tDígame el número de componentes del vector: ");  
    scanf ("%d", &n);  
    while ( n <= 0 )  
    { printf ("\n\t\t\tEl número de componentes debe ser entero, positivo y menor que %d, digite  
nuevamente: ", nmax);  
        scanf ("%d", &n); }  
    printf ("\n");  
    for ( i = 0; i < n; i++ )  
    { printf ("\t\t\t\tDígame la componente %d del vector: ", i);  
        scanf ("%f", &v[i]); }  
    printf ("\n\t\tDígame la posición (0 <= m <= %d) y la componente a eliminar del vector: ", n);  
    scanf ("%d", &m);  
    while ( m < 0 || m > n )  
    { printf ("\n\t\t\t\tPosición de la componente a eliminar en el vector incorrecta, digite nuevamente: ");  
        scanf ("%d", &m); }  
    for ( i = m; i < n - 1; i++ )  
    { v[i] = v[i+1]; }  
    printf("\n");  
    for ( i = 0; i < n - 1; i++ )  
    { printf ("\n\t\t\t\tLa componente %d del vector es %f: ", i, v[i]); }  
    printf("\n\n");  
}  
void cls(void)  
{  
    printf ("\033[2J"); /* Limpia la pantalla */  
    printf ("\033[0;0f"); /* Mueve el cursor en la esquina superior izquierda */  
}
```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VECTORES EN LENGUAJE C

/* Vect07: Escribir un programa en lenguaje C que:
Permita determinar el máximo común divisor y el mínimo común múltiplo de un conjunto de n números enteros y
positivos
dados. */

```
#define nmax 50
#include <stdio.h>
void cls(void);
int McdDos(int a, int b);
main()
{
    int i, n, a, b, mcd, mcm;
    int nros[nmax];
    cls();
    printf ("\n\n\t Digite cuantos números enteros y positivos considerará ");
    scanf ("%d", &n);
    while ( n <= 1 || n > nmax )
    { printf("\n\t El número de datos debe ser mayor que uno y menor que %d, digite nuevamente, ", nmax);
        scanf("%d", &n); }
    printf("\n");
    for (i = 0; i < n; i++)
    { printf ("\t\t\t Digite nro( %d ) = ", i);
        scanf("%d", &nros[i]);
        while (nros[i] <= 0 )
        { printf("El número debe ser positivo, digite nuevamente, ");
            scanf("%d", &nros[i]); } }
    a = nros[0];
    b = nros[1];
    mcd = McdDos(a, b);
    mcm = a*b/mcd;
    for (i = 2; i < n; i++)
    { a = mcd;
        b = nros[i];
        mcd = McdDos(a, b); }
    for (i = 2; i < n; i++)
    { a = mcm;
        b = nros[i];
        mcm = a*b/McdDos(a, b); }
    printf("\n\n\t El mcd y el mcm de los números { %d", nros[0]);
    for ( i = 1; i < n; i++)
    { printf(", %d ", nros[i]); }
    printf(" } son %d, %d, respectivamente \n\n", mcd, mcm);
}
void cls(void)
{
    printf("\033[2J");           /* Limpia la pantalla */
    printf("\033[0;0f");         /* Mueve el cursor en la esquina superior izquierda */
}
int McdDos(int a, int b)
{
    int p, q, r;
    if (a < b)
    { p = b;
        q = a; }
    else
    { p = a;
        q = b; }
    r = p%q;
    while ( r != 0 )
    { p = q;
        q = r;
        r = p%q; }
    return(q);
}
```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VECTORES EN LENGUAJE C

```
/* Vect08: Escribir un programa en lenguaje C que:  
Lea un número entero n, que si es positivo lo acepte; en caso contrario, solicite de nuevo n hasta  
obtener uno  
que sea positivo. Lea luego las n componentes del vector v y determine otro vector u cuyas  
componentes son,  
respectivamente, el valor que toma la función y = 3x2 - 2x + 4, para cada componente del vector v.  
*/  
#define nmax 50  
#include <stdio.h>  
void cls(void);  
main()  
{  
    int i, n;  
    float u[nmax], v[nmax];  
    cls();  
    printf ("\n\t\t\t\tDigite el número de componentes del vector: ");  
    scanf ("%d", &n);  
    while ( n <= 0 )  
    { printf ("\n\t\t\t\tEl número de componentes debe ser entero, positivo y menor que %d, digite  
nuevamente: ", nmax);  
        scanf ("%d", &n); }  
    printf ("\n");  
    for ( i = 0; i < n; i++ )  
    { printf ("\t\t\t\tDigite la componente %d del vector: ", i);  
        scanf ("%f", &v[i]); }  
    for ( i = 0; i < n; i++ )  
    { u[i] = 3*v[i]*v[i] - 2*v[i] + 4; }  
    printf("\n\t Vector dado (v) - Vector obtenido\n\t\t\t ( u[i] = 3*v[i]*v[i] - 2*v[i] + 4 )");  
    for ( i = 0; i < n; i++ )  
    { printf (" \n\t\t%10f\t\t\t%10f", v[i], u[i]); }  
    printf("\n\n");}  
void cls(void)  
{  
    printf ("\033[2J"); /* Limpia la pantalla */  
    printf ("\033[0;0f"); /* Mueve el cursor en la esquina superior izquierda */}
```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VECTORES EN LENGUAJE C

/* **Vect09:** Escribir un programa en lenguaje C que:
Lea un número entero n, que si es positivo lo acepte; en caso contrario, solicite de nuevo n hasta
obtener un
número positivo. Lea luego las n componentes del vector vect y determine si existen componentes
consecutivas
iguales. En el caso que existan, imprima dichas componentes consecutivas y las posiciones que
ocupan. Si no existen
componentes consecutivas iguales imprima el mensaje El vector no tiene componentes consecutivas
iguales. */

```
#define nmax 50
#include <stdio.h>
void cls(void);
main()
{
    int i, j, n;
    float vect[nmax];
    cls();
    printf ("\n\t\t\t\t\tDigite el número de componentes del vector: ");
    scanf ("%d", &n);
    while ( n <= 0 )
    { printf ("\n\t\t\t\tEl número de componentes debe ser entero, positivo y menor que %d, digite
nuevamente: ", nmax);
        scanf ("%d", &n); }
    printf ("\n");
    for ( i = 0; i < n; i++ )
    { printf ("\t\t\t\t\tDigite la componente %d del vector: ", i);
        scanf ("%f", &vect[i]); }
    i = 0;
    while ( i < n - 1 )
    { j = i + 1;
        if ( vect[j] == vect[i] )
        { printf ("\n\t\t\tLa componente %10f ocupa las posiciones: %3d ", vect[i], i); }
        while ( vect[j] == vect[i] )
        { printf (",%3d ", j);
            j++; }
        i = j; }
    printf ("\n\n");
}
void cls(void)
{
    printf ("\033[2J");      /* Limpia la pantalla          */
    printf ("\033[0;0f");    /* Mueve el cursor en la esquina superior izquierda */
}
```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VECTORES EN LENGUAJE C

```
/* Vect10: Escribir un programa en lenguaje C que:  
Lea un número entero n, que si es positivo lo acepte; en caso contrario, solicite de nuevo n hasta  
obtener uno que  
sea positivo. Lea luego las n componentes del vector vect y determine si existen componentes del  
vector vect que  
sean iguales al cuadrado de alguna de las demás. Si existen dichas componentes, imprimirlo, en cada  
caso, con aquella  
cuyo cuadrado le es igual, como también las posiciones que ocupan. Si ninguna de las componentes  
resulta igual al  
cuadrado de alguna de las demás, imprima el mensaje "Ninguna de las componentes del vector es igual  
al cuadrado de las  
demás". */  
#define nmax 50  
#include <stdio.h>  
void cls(void);  
main()  
{  
    int i, j, l, lg, n;  
    float vect[nmax];  
    cls();  
    printf ("\n\t\t\t\t\tDigite el número de componentes del vector: ");  
    scanf ("%d", &n);  
    while ( n <= 0 )  
    { printf ("\n\t\t\t\tEl número de componentes debe ser entero, positivo y menor que %d, digite  
nuevamente: ", nmax);  
        scanf ("%d", &n); }  
    printf ("\n");  
    for ( i = 0; i < n; i++ )  
    { printf ("\t\t\t\t\tDigite la componente %d del vector: ", i);  
        scanf ("%f", &vect[i]); }  
    lg = 0;  
    for ( i = 0; i < n; i++ )  
    { l = 0;  
        for ( j = 0; j < n; j++ )  
        { if ( j != i )  
            { if ( vect[i] == vect[j]*vect[j] )  
                { if ( l == 0 )  
                    { l = 1;  
                        printf ("\n\t\t\tLa componente vect[%d] = %10f es igual al cuadrado de la componente %f en: %  
3d", i, vect[i], vect[j], j);  
                        lg = 1; }  
                    else  
                        { printf (" , %3d", j); } } } } }  
    if ( lg == 0 )  
    { printf ("\nNo existen componentes que son iguales a los cuadrados de los demás"); }  
    printf ("\n\n");}  
void cls(void)  
{  
    printf ("\033[2J"); /* Limpia la pantalla */  
    printf ("\033[0;0f"); /* Mueve el cursor en la esquina superior izquierda */}
```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VECTORES EN LENGUAJE C

```
/* Vect11: Escribir un programa en lenguaje C que:  
Lea un número entero n, que si es positivo lo acepte; en caso contrario, solicite de nuevo n hasta  
obtener uno que  
sea positivo. Lea luego las n componentes del vector v y determine otro vector u de n - 1  
componentes, obtenidas  
como la suma de las componentes que ocupan posiciones consecutivas en v, siendo la primera  
componente de u la suma de  
la primera componente de v con la segunda y así sucesivamente. */  
#define nmax 50  
#include <stdio.h>  
void cls(void);  
main()  
{  
    int i, j, n;  
    float v[nmax], u[nmax - 1];  
    cls();  
    printf ("\n\t\t\t\t\tDigue el número de componentes del vector: ");  
    scanf ("%d", &n);  
    while ( n <= 0 )  
    { printf ("\n\t\t\tEl número de componentes debe ser entero, positivo y menor que %d, digite  
nuevamente: ", nmax);  
        scanf ("%d", &n); }  
    printf ("\n");  
    for ( i = 0; i < n; i++ )  
    { printf ("\t\t\t\tDigue la componente %d del vector: ", i);  
        scanf ("%f", &v[i]); }  
    j = -1;  
    for ( i = 0; i < n - 1; i = i + 1 )  
    { j++;  
        u[j] = v[i] + v[i + 1]; }  
    printf("\n\t Vector dado (v)      -      Vector obtenido\n\t\t\t\t( u[i] = v[i] + v[i + 1])");  
    for ( i = 0; i < n - 1; i++ )  
    { printf (" \n\t\t\t%10f\t\t\t%10f", v[i], u[i] ); }  
    printf("\n\t\t\t%10f\n", v[n - 1]);  
}  
void cls(void)  
{  
    printf ("\033[2J");      /* Limpia la pantalla */  
    printf ("\033[0;0f");     /* Mueve el cursor en la esquina superior izquierda */  
}
```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VECTORES EN LENGUAJE C

```

/* Vect12a: Escribir un programa en lenguaje C que:
Permita determinar si un número a se encuentra entre las n componentes de un vector. Resolver el
problema utilizando
la búsqueda secuencial. */
#define nmax 50
#include <stdio.h>
void cls(void);
main()
{
    int i, n, lg;
    float a, v[nmax];
    cls();
    printf ("\n\t\t\t\t\t\tDigite el número de componentes del vector: ");
    scanf ("%d", &n);
    while ( n <= 0 )
    { printf ("\n\t\t\t\tEl número de componentes debe ser entero, positivo y menor que %d, digite
nuevamente: ", nmax);
        scanf ("%d", &n); }
    printf ("\n");
    for ( i = 0; i < n; i++ )
    { printf ("\t\t\t\t\t\tDigite la componente %d del vector: ", i);
        scanf ("%f", &v[i]); }
    printf ("\n\t\t\t\t\t\tDigite el elemento que buscará en el vector: ");
    scanf ("%f", &a);
    lg = 0;
    i = -1;
    while ( i < n - 1 && lg == 0 )
    { i++;
        if ( v[i] == a )
        { lg = 1; } }
    if ( lg == 1 )
    { printf ("\n\t\t\t\t\t\tEl elemento dado ocupa la posición %d en el vector", i); }
    else
    { printf ("\n\t\t\t\t\t\tEl elemento dado no se encuentra en el vector"); }
    printf("\n\n");
}
void cls(void)
{
    printf ("\033[2J"); /* Limpia la pantalla */
    printf ("\033[0;0f"); /* Mueve el cursor en la esquina superior izquierda */
}

```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VECTORES EN LENGUAJE C

```

/* Vect12b: Escribir un programa en lenguaje C que:
Permita determinar si un número a se encuentra entre las n componentes de un vector. Resolver el
problema utilizando
la búsqueda binaria12. Recordar que para aplicar la búsqueda binaria, las componentes deben estar
ordenadas. */
#define nmax 50
#include <stdio.h>
void cls(void);
main()
{
    int i, j, k, n, lg;
    float a, v[nmax];
    cls();
    printf ("\n\tDigite el número de componentes del vector, cuyas componentes deben estar en secuencia
ascendente : ");
    scanf ("%d", &n);
    while ( n <= 0 )
    { printf ("\n\t\tEl número de componentes debe ser entero, positivo y menor que %d, digite
nuevamente: ", nmax);
        scanf ("%d", &n); }
    printf ("\n\t\tDigite la componente 0 del vector: ");
    scanf ("%f", &v[0]);
    for ( i = 1; i < n; i++ )
    { printf ("\t\t\tDigite la componente %d del vector: ", i);
        scanf ("%f", &v[i]);
        while ( v[i] < v[i - 1] )
        { printf ("\tLas componentes deben estar en secuencia ascendente, digite nuevamente el elemento v
[%d]", i);
            scanf ("%f", &v[i]); } }
    printf ("\n\t\tDigite el elemento que buscará en el vector: ");
    scanf ("%f", &a);
    lg = 0;
    i = 0;
    j = n - 1;
    while ( i < j && lg == 0 )
    { k = ( i + j ) / 2;
        if ( a == v[k] )
        { lg = 1; }
        else
        { if ( a < v[k] )
            { j = k - 1; }
            else
            { i = k + 1; } } }
    if ( lg == 1 )
    { printf ("\n\t\t\tEl elemento dado ocupa la posición %d en el vector", k); }
    else
    { printf ("\n\t\t\tEl elemento dado no se encuentra en el vector"); }
    printf ("\n\n");
}
void cls(void)
{
    printf ("\033[2J"); /* Limpia la pantalla */
    printf ("\033[0;0f"); /* Mueve el cursor en la esquina superior izquierda */
}

```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VECTORES EN LENGUAJE C

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VECTORES EN LENGUAJE C

/* **Vect13b:** Escribir un programa en lenguaje C que:
 Permite ordenar, en orden ascendente, las n componentes de un vector de componentes numéricas, utilizando el método de la burbuja. */

```
#define nmax 50
#include <stdio.h>
void cls(void);
main()
{
    int i, j, l, n, lg;
    float a, v[nmax];
    cls();
    printf ("\n\tDigite el número de componentes del vector, cuyas componentes quiere ordenar en secuencia ascendente : ");
    scanf ("%d", &n);
    while ( n <= 0 )
    { printf ("\n\t\tEl número de componentes debe ser entero, positivo y menor que %d, digite nuevamente: ", nmax);
        scanf ("%d", &n); }
    printf ("\n");
    for ( i = 0; i < n; i++ )
    { printf ("\t\t\t\tDigite la componente %d del vector: ", i);
        scanf ("%f", &v[i]); }
    lg = 0;
    l = n;
    while ( lg == 0 )
    { lg = 1;
        l--;
        for ( j = 0; j < l; j++ )
        { if ( v[j] > v[j + 1] )
            { lg = 0;
                a = v[j];
                v[j] = v[j + 1];
                v[j + 1] = a; } } }
    printf ("\n\tComponentes ordenados { %10f", v[0]);
    for ( i = 1; i < n; i++ )
    { if ( i%4 == 0 )
        { printf ("\n\t\t\t\t%10f", v[i]); }
        else
        { printf (" , %10f", v[i]); } }
    printf(" }\n\n");
}
void cls(void)
{
    printf ("\033[2J");      /* Limpia la pantalla */
    printf ("\033[0;0f");    /* Mueve el cursor en la esquina superior izquierda */
}
```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VECTORES EN LENGUAJE C

```
/* Vect13c: Escribir un programa en lenguaje C que:  
Permita ordenar, en orden ascendente, las n componentes de un vector de componentes numéricas,  
utilizando la función qsort. */  
#include <stdlib.h>  
#include <stdio.h>  
#define nmax 50  
void mostrar(float *, int);  
int comp(const void *, const void *);  
void cls(void);  
main()  
{  
    float vect[nmax];           /* Números a ser ordenados. */  
    int i, n;                   /* Número de datos. */  
    cls();  
    printf ("\n\t\t\t\tDigite el número de componentes del vector: ");  
    scanf ("%d", &n);  
    while ( n <= 0 || n > nmax )  
    { printf ("\n\t\t\t\tEl número de componentes del vector debe ser entero, positivo y no mayor que %d,  
digite nuevamente: ", nmax);  
        scanf ("%d", &n); }  
    printf ("\n");  
    for (i = 0; i < n; i++)  
    { printf ("\t\t\t\t\t\tDigite vect( %d ) = ", i);  
        scanf("%f", &vect[i]); }  
    puts("\nEstos son los números dados");  
    mostrar(vect, n);  
    qsort(                /* Ordena los números contenidos en vect */  
          vect,            /* Puntero a los elementos */  
          n,               /* Numero de elementos */  
          sizeof(float),   /* Ancho de cada elemento */  
          comp             /* Puntero a la función de comparación */ );  
    puts("\nEstos son los números ordenados");  
    mostrar(vect, n);  
    printf("\n");}  
// función comparar: Compara dos números.  
int comp(const void *nro1, const void *nro2)  
{  
    return *(float *)nro1 - *(float *)nro2;  
}  
// función mostrar: Despliega en pantalla los números.  
void mostrar(float *vector, int nro)  
{  
    printf("\t");  
    while ( nro-- )  
    {  
        printf("%f ",*vector);  
        vector++;  
    }  
    puts("");}  
void cls(void)  
{  
    printf ("\033[2J");      /* Limpia la pantalla */  
    printf ("\033[0;0f");     /* Mueve el cursor en la esquina superior izquierda */}
```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VECTORES EN LENGUAJE C

```

/*
 * Vect13d: Escribir un programa en lenguaje C que:
 * Permite ordenar, en orden ascendente, las n componentes de un vector de componentes numéricas,
 * utilizando el método de ordenación rápida (quicksort) */
#include <stdlib.h>
#include <stdio.h>
#define nmax 50
void mostrar(float *, int);
void cls(void);
main()
{
    float vect[nmax], vectaux[nmax], aux, piv;           /* Números enteros a ser ordenados.    */
    int i, j, n, k, l, p, q, ij[nmax][2];                /* Número de datos.                  */
    cls();
    printf ("\n\t\t\t\t\tDigue el número de componentes del vector: ");
    scanf ("%d", &n);
    while ( n <= 0 || n > nmax )
    { printf ("\n\t\t\t\t\tEl numero de componentes debe ser entero, positivo y no mayor que %d, digite nuevamente: ", nmax);
        scanf ("%d", &n); }
    printf ("\n");
    for (i = 0; i < n; i++)
    { printf ("\t\t\t\t\t\t\tDigue vect( %d ) = ", i);
        scanf("%f", &vect[i]); }
    puts("\n\tEstos son los números dados");
    mostrar(vect, n);
    for (i = 0; i < n; i++)
    { vectaux[i] = vect[i]; }
    l = 0;   ij[l][0] = 0;   ij[l][1] = n - 1;
    while ( l > -1 )
    { i = ij[l][0];
        j = ij[l][1];   l = l - 1;
        while ( i < j )
        { p = i;
            q = j;   piv = vectaux[p];   p++;
            while ( p <= q )
            { while ( vectaux[p] <= piv && p <= q )
                { p++; }
                while ( vectaux[q] > piv && p <= q )
                { q--; }
                if ( p <= q )
                { aux = vectaux[p];
                    vectaux[p] = vectaux[q];   vectaux[q] = aux;
                    p++;   q--; } } }
        aux = vectaux[i];
        vectaux[i] = vectaux[q];
        vectaux[q] = aux;
        if ( q - i < j - q )
        { if ( q < j )
            { l++;
                ij[l][0] = q + 1;
                ij[l][1] = j; }
            j = q - 1; }
        else
        { if ( i < q )
            { l++;
                ij[l][0] = i;
                ij[l][1] = q - 1; }
            i = q + 1; } } }
    puts ("\tEstos son los números ordenados");
    mostrar(vectaux, n);
}
// función mostrar: Despliega en pantalla los números.
void mostrar(float *vector, int nro)
{
    printf("\t");
    while ( nro-- )
    { printf("%f ", *vector);
        vector++; }
    puts("\n");
}
// función cls: limpia la pantalla.
void cls(void)
{
    printf ("\033[2J");      /* Limpia la pantalla               */
    printf ("\033[0;0f");    /* Mueve el cursor en la esquina superior izquierda */
}

```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO MATRICES EN LENGUAJE C

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO MATRICES EN LENGUAJE C

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO MATRICES EN LENGUAJE C

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO MATRICES EN LENGUAJE C

/* **Matr04:** Escribir un programa en lenguaje C que:

Lea una matriz cuadrada de orden n , e imprima dos matrices que sumadas dan la matriz dada, y tal que una de ellas sea

simétrica y la otra antisimétrica. */

```
#define nmax 50
#include <stdio.h>
void cls(void);
main()
{
    int i, j, k, n;
    float mat[nmax][nmax], msim[nmax][nmax], masim[nmax][nmax];
    cls();
    printf ("\n\t\t\t\t\tDigite el orden de la matriz: ");
    scanf ("%d", &n);
    while ( n <= 0 || n > nmax )
    { printf ("\n\t\t\t\t\tEl orden de la matriz debe ser entero, positivo y no mayor que %d, digite nuevamente: ", nmax);
        scanf ("%d", &n);
    }
    printf ("\n");
    for ( i = 0; i < n; i++ )
    { for ( j = 0; j < n; j++ )
        { printf ("\t\t\t\t\tDigite la componente mat[%d][%d]: ", i, j);
            scanf ("%f", &mat[i][j]); }
    }
    for ( i = 0; i < n; i++ )
    { for ( j = 0; j < n; j++ )
        { msim[i][j] = (mat[i][j] + mat[j][i])/2;
            masim[i][j] = (mat[i][j] - mat[j][i])/2; } }
    cls();
    printf("\n\t\t\t\t\tOrden de las matrices %3d", n);
    printf("\n\t\t\t\t\tMatriz dada");
    for ( i = 0; i < n; i++ )
    { if ( i == 0 )
        { printf ("\n{ ");
        }
        else
        { printf ("\n {");
        }
        for ( j = 0; j < n; j++ )
        { if ( j < n - 1 )
            { printf ("%10f, ", mat[i][j]); }
            else
            { printf ("%10f }", mat[i][j]); } }
    }
    printf (" }\n\n\t\t\t\t\tMatriz simétrica");
    for ( i = 0; i < n; i++ )
    { if ( i == 0 )
        { printf ("\n{ ");
        }
        else
        { printf ("\n {");
        }
        for ( j = 0; j < n; j++ )
        { if ( j < n - 1 )
            { printf ("%10f, ", msim[i][j]); }
            else
            { printf ("%10f }", msim[i][j]); } }
    }
    printf (" }\n\n\t\t\t\t\tMatriz antisimétrica");
    for ( i = 0; i < n; i++ )
    { if ( i == 0 )
        { printf ("\n{ ");
        }
        else
        { printf ("\n {");
        }
        for ( j = 0; j < n; j++ )
        { if ( j < n - 1 )
            { printf ("%10f, ", masim[i][j]); }
            else
            { printf ("%10f }", masim[i][j]); } }
    }
    printf (" }\n\n");
}
void cls(void)
{
    printf ("\033[2J");      /* Limpia la pantalla          */
    printf ("\033[0;0f");    /* Mueve el cursor en la esquina superior izquierda */
}
```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO MATRICES EN LENGUAJE C

```

/* Matr05: Escribir un programa en lenguaje C que:
   Permita determinar la traza de una matriz cuadrada de orden n. */
#define nmax 50
#include <stdio.h>
void cls(void);
main()
{
    int i, j, n;
    float traza, mat[nmax][nmax];
    cls();
    printf ("\n\t\t\t\t\t\t\t\t\tDigite el orden de la matriz: ");
    scanf ("%d", &n);
    while ( n <= 0 || n > nmax )
    { printf ("\n\t\t\t\t\t\t\t\t\tEl orden de la matriz debe ser entero, positivo y no mayor que %d, digite nuevamente: ", nmax);
        scanf ("%d", &n); }
    printf ("\n");
    for ( i = 0; i < n; i++ )
    { for ( j = 0; j < n; j++ )
        { printf ("\t\t\t\t\t\t\t\t\tDigite la componente mat[%d][%d]: ", i, j);
            scanf ("%f", &mat[i][j]); } }
    traza = 0;
    for ( i = 0; i < n; i++ )
    { traza += mat[i][i]; }
    cls();
    printf ("\n\n\t\t\t\t\t\t\t\tMatriz dada, de orden %3d", n);
    for ( i = 0; i < n; i++ )
    { if ( i == 0 )
        { printf ("\n{ "); }
        else
        { printf ("\n {"); }
        for ( j = 0; j < n; j++ )
        { if ( j < n - 1 )
            { printf ("10f, ", mat[i][j]); }
            else
            { printf ("%10f ", mat[i][j]); } } }
    printf (" }\n\t\t\t\t\t\t\t\t\tTraza de la matriz dada %f\n\n", traza);
}
void cls(void)
{
    printf ("\033[2J"); /* Limpia la pantalla */
    printf ("\033[0;0f"); /* Mueve el cursor en la esquina superior izquierda */
}

```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO MATRICES EN LENGUAJE C

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO MATRICES EN LENGUAJE C

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO MATRICES EN LENGUAJE C

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO MATRICES EN LENGUAJE C

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO MATRICES EN LENGUAJE C

/* **Matr10:** Escribir un programa en lenguaje C que:

Lea dos números enteros y positivos m y n , y una matriz mat , de componentes numéricas, de orden $m \times n$ e imprima la matriz ampliada con: una fila que contenga la suma de los elementos de las columnas correspondientes, una columna que contenga la suma de los elementos de las filas correspondientes, y el elemento $(m + 1) \times (n + 1)$ sea igual a la suma de todos los elementos de la matriz. */

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO MATRICES EN LENGUAJE C

```
/* Matr11: Escribir un programa en lenguaje C que:  
Permita resolver un sistema de ecuaciones lineales de n ecuaciones con n incógnitas, sabiendo que  
la matriz del  
sistema es triangular superior. */  
#define nmax 50  
#include <stdio.h>  
void cls(void);  
main()  
{  
    int i, j, n;  
    float a[nmax][nmax], b[nmax], x[nmax];  
    cls();  
    printf ("\n\t\t\t\t\tDigite el número de ecuaciones del sistema: ");  
    scanf ("%d", &n);  
    while ( n <= 0 || n > nmax )  
    { printf ("\n\t\t\tEl número de ecuaciones debe ser entero, positivo y no mayor que %d, digite  
nuevamente: ", nmax);  
        scanf ("%d", &n); }  
    printf ("\n\t\t\t\tDigite los coeficientes de la ecuación, seguido del término independiente");  
    for ( i = 0; i < n; i++ )  
    { printf ("\n");  
        for ( j = 0; j < n; j++ )  
        { printf ("\t\t\tCoeficiente a[%d][%d]: ", i, j);  
            scanf ("%f", &a[i][j]);  
            while ( i > j && a[i][j] != 0 )  
            { printf ("\n\t\t\t\t\tEsta componente debe ser cero, Digite nuevamente a[%d][%d]: ", i, j);  
                scanf ("%f", &a[i][j]); }  
            while ( i == j && a[i][j] == 0 )  
            { printf ("\n\t\t\t\t\tEsta componente no puebe ser cero, Digite nuevamente a[%d][%d]: ", i, j);  
                scanf ("%f", &a[i][j]); } }  
        printf ("\t\t\tTérmino independiente b[%d]: ", i);  
        scanf ("%f", &b[i]); }  
    x[n - 1] = b[n - 1]/a[n - 1][n - 1];  
    for ( i = n - 2; i >= 0; i-- )  
    { x[i] = b[i];  
        for ( j = i + 1; j < n; j++ )  
        { x[i] -= a[i][j]*x[j]; }  
        x[i] = x[i]/a[i][i]; }  
    cls();  
    printf ("\n\n\tSistema de %d ecuaciones dado", n);  
    for ( i = 0; i < n; i++ )  
    { printf ("\n");  
        for ( j = 0; j < n; j++ )  
        { printf ("%10f ", a[i][j]); }  
        printf (" %10f", b[i]); }  
    printf ("\n\n\tSolución del sistema = { %10f", x[0]);  
    for ( i = 1; i < n; i++ )  
    { printf (", %10f", x[i]); }  
    printf (" }\n\n"); }  
void cls(void)  
{  
    printf ("\033[2J"); /* Limpia la pantalla */  
    printf ("\033[0;0f"); /* Mueve el cursor en la esquina superior izquierda */ } }
```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO MATRICES EN LENGUAJE C

```
/* Matr12: Escribir un programa en lenguaje C que:  
   Permita resolver un sistema de ecuaciones lineales de m ecuaciones con n incógnitas. */  
#define nmax 50  
#include <stdio.h>  
void cls(void);  
main()  
{  
    int f, c, i, j, k, l, m, n, p, lg, fil[nmax], col[nmax];  
    float det, a[nmax][nmax], aux[nmax][nmax], b[nmax], auxx[nmax];  
// Lectura de datos  
    cls();  
    printf("\n\n\t\t\tSolución de un sistema lineal de m ecuaciones con n incógnitas\n");  
// Lectura de datos.  
    printf("\n\t\tDigitar el número de ecuaciones m y el número n de incógnitas: ");  
    scanf("%d %d", &m, &n);  
    while ( m <= 0 || m > nmax || n <= 0 || n > nmax )  
    { printf("\tEl número de ecuaciones y de incógnitas debe ser entero, positivo y menor que %d, digite nuevamente: ",  
nmax);  
        scanf("%d %d", &m, &n); }  
    for ( i = 0; i < m; i++ )  
    { printf("\n");  
        for ( j = 0; j < n; j++ )  
        { printf("\t\t\tDigitar el coeficiente a[%d][%d]: ", i, j);  
            scanf("%f", &a[i][j]);  
            auxx[i][j] = a[i][j]; }  
        printf("\t\t\tDigitar el término independiente b[%d]: ", i);  
        scanf("%f", &b[i]);  
        aux[i] = b[i];  
        fil[i] = -1;  
        col[i] = -1; }  
    if ( m == 1 && n == 1 )  
    { if ( a[m - 1][n - 1] != 0 )  
        { b[m - 1] = b[m - 1]/a[m - 1][n - 1];  
            printf ("\n\n\t\t\tEs una ecuación con una incógnita: x = %f\n\n", b[m - 1]); }  
        else  
        { if ( b[m - 1] != 0 )  
            { printf ("\n\n\t\t\tEs una ecuación con una incógnita, pero incompatible\n\n"); }  
            else  
            { printf ("\n\n\t\t\tEs una ecuación con una incógnita, pero indeterminado\n\n"); } } }  
    else  
    {  
// Búsqueda del pivot a[f][c]. lg = 1 cuando encuentra un pivot.  
        l = -1;  
        for ( c = 0; c < n; c++ )  
        { f = -1;  
            lg = 0;  
            while ( f < m - 1 && lg == 0 )  
            { f++;  
                if ( a[f][c] != 0 )  
                { k = 0;  
                    while ( k <= l && fil[k] != f )  
                    { k++; }  
                    if ( k > l )  
                    { lg = 1;  
                        l++;  
                        fil[l] = f;  
                        col[l] = c; } } } }  
// Triangularización. lg = 1 si encontró un pivot.  
    if ( lg == 1 )  
    { for ( i = 0; i < m; i++ )  
        { k = 0;  
            while ( k <= l && fil[k] != i )  
            { k++; }  
            if ( k > l )  
            { for ( j = c + 1; j < n; j++ )  
                { a[i][j] = a[i][j]*a[f][c] - a[i][c]*a[f][j]; }  
                b[i] = b[i]*a[f][c] - b[f]*a[i][c];  
                a[i][c] = 0; } } } }
```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO MATRICES EN LENGUAJE C

```
// Matr12: (Cont.) Impresión del sistema de ecuaciones dado.
cls();
printf ("\n\n\n\tSistema de %d ecuaciones con %d incógnitas dado", m, n);
for ( i = 0; i < m; i++ )
{ printf ("\n");
  for ( j = 0; j < n; j++ )
  { printf ("%10f ", aaux[i][j]); }
  printf (" %10f", baux[i]); }
// Impresión del sistema con matriz en forma escalonada.
printf("\n\n\n\tSistema equivalente en forma escalonada");
for ( i = 0; i < m; i++ )
{ printf ("\n");
  for ( j = 0; j < n; j++ )
  { printf("%10f", a[i][j]); }
  printf("%10f", b[i]); }
// Análisis si tiene infinitas soluciones (lg = 0) o si es incompatible (lg = 1).
lg = 0;
for ( i = 0; i < m; i++ )
{ k = 0;
  while ( k <= l && fil[k] != i )
  { k++; }
  if ( k > l )
  { j = 0;
    while ( j < n && a[i][j] == 0 )
    { j++; }
    if ( j == n && b[i] != 0 ) lg = 1; } }
if ( lg == 1 )
{ printf("\n\n\tEl sistema dado es incompatible\n\n"); }
else
// Paso a la forma canónica.
{ for ( k = m - 1; k >= 0; k-- )
  { f = fil[k];
    c = col[k];
    if ( c != -1 )
    { for ( j = c + 1; j < n; j++ )
      { af[j] = a[f][j]/a[f][c]; }
    b[f] = b[f]/a[f][c];
    a[f][c] = 1;
    for ( p = k - 1; p >= 0; p-- )
    { i = fil[p];
      for ( j = c + 1; j < n; j++ )
      { a[i][j] = a[i][j] - a[i][c]*a[f][j]; }
      b[i] = b[i] - a[i][c]*b[f];
      a[i][c] = 0; } } }
// Impresión del sistema con matriz en forma canónica.
printf("\n\n\n\tSistema equivalente en forma canónica");
for ( i = 0; i < m; i++ )
{ printf("\n");
  for ( j = 0; j < n; j++ )
  { printf("%10f", a[i][j]); }
  printf("%10f", b[i]); }
if ( l == n )
// Impresión de la solución del sistema.
{ i = 0;
  printf ("\n\n\n\tSolución del sistema = { x%-3d = %10f", i, b[i]);
  for ( i = 1; i < n; i++ )
  { printf (" , x%-3d = %10f", i, b[i]); }
  printf (" }\n\n"); }
else
{ printf("\n\n\tEl sistema admite infinitas soluciones\n\n"); }
}
void cls(void)
{
  printf ("\033[2J"); /* Limpia la pantalla */
  printf ("\033[0;0f"); /* Mueve el cursor en la esquina superior izquierda */
}
```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO MATRICES EN LENGUAJE C

```

/*
Matr13: Escribir un programa en lenguaje C que:
    Permita calcular el determinante de una matriz cuadrada de orden n. */
#define nmax 50
#include <stdio.h>
void cls(void);
main()
{
    int f, c, i, j, k, l, n, lg, fil[nmax];
    float det, mat[nmax][nmax];
// Lectura de datos
cls();
printf("\n\n\t\t\t\t\tDeterminante de una matriz de orden n\n");
printf("\n\t\t\t\t\tDigitar el orden de la matriz: ");
scanf("%d", &n);
printf("\n");
while ( n <= 0 )
{ printf("\t\tEl orden de la matriz debe ser entero, positivo y menor que %d, digite nuevamente: ", nmax);
    scanf("%d", &n); }
for ( i = 0; i < n; i++ )
{ printf("\n");
    for (j = 0; j < n; j++ )
    { printf("\t\t\tDigitar la componente mat[%d][%d]: ", i, j);
        scanf("%f", &mat[i][j]); }
    fil[i] = 0; }
// Impresión de la matriz dada
cls();
printf("\n\t\tMatriz dada (se hallará su determinante)\n\n");
for ( i = 0; i < n; i++ )
{ for ( j = 0; j < n; j++ )
    { printf("\t%6f", mat[i][j]); }
    printf("\n"); }
// Búsqueda del pivot y triangularización
l = -1;
det = 1;
for ( c = 0; c < n; c++ )
{ lg = 0;
    f = -1;
// Búsqueda del pivot mat[f][c]. (lg == 1 cuando se encuentra el pivot)
    while ( f < n - 1 && lg == 0 )
    { f++;
        if ( mat[f][c] != 0 )
        { k = 0;
            while ( k <= l && fil[k] != f )
            { k++; }
            if ( k > l )
            { lg = 1;
                l++;
                fil[l] = f; } } } }
// Triangularización
if ( lg == 1 )
{ for ( i = 0; i < n; i++ )
    { k = 0;
        while ( k <= l && fil[k] != i )
        { k++; }
        if ( k > l )
        { det = det/mat[f][c];
            for ( j = c + 1; j < n; j++ )
            { mat[i][j] = mat[i][j]*mat[f][c] - mat[i][c]*mat[f][j]; }
            mat[i][c] = 0; } } } }
// Completar el cálculo del determinante de la matriz ya triangularizada
if ( l < n - 1 )
{ det = 0; }
else
{ for ( i = 0; i < n; i++ )
    { c = fil[i];
        det = det*mat[c][c]; } }
// Impresión del determinante de la matriz
printf("\nEl determinante de la matriz es: %f\n", det);
}
void cls(void)
{
    printf("\033[2J"); /* Limpia la pantalla */
    printf("\033[0;0f"); /* Mueve el cursor en la esquina superior izquierda */
}

```

/* **Matr14:** Escribir un programa en lenguaje C que:
Permita invertir una matriz cuadrada de orden n. */

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO MATRICES EN LENGUAJE C

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO MATRICES EN LENGUAJE C

```
// Matr14: (Cont.) Impresión de la matriz dada.  
cls();  
printf ("\n\n\t\t\t\tMatriz dada");  
for ( i = 0; i < n; i++ )  
{ printf ("\n");  
    for ( j = 0; j < n; j++ )  
    { printf ("%10f ", aaux[i][j]); } }  
// Impresión de la matriz en forma escalonada.  
printf("\n\n\t\tMatriz equivalente en forma escalonada");  
for ( i = 0; i < n; i++ )  
{ printf ("\n");  
    for ( j = 0; j < n; j++ )  
    { printf("%10f", a[i][j]); } }  
// Análisis si la matriz es invertible.  
if ( l < n - 1 )  
{ printf("\n\n\t\tLa matriz dada no es invertible\n\n"); }  
else  
// Paso a la forma canónica.  
{ for ( k = n - 1; k >= 0; k-- )  
{ f = fil[k];  
    c = col[k];  
    for ( j = c + 1; j < n; j++ )  
    { a[f][j] = a[f][j]/a[f][c]; }  
    for ( j = 0; j < n; j++ )  
    { b[f][j] = b[f][j]/a[f][c]; }  
    a[f][c] = 1;  
    for ( p = k - 1; p >= 0; p-- )  
    { i = fil[p];  
        for ( j = c + 1; j < n; j++ )  
        { a[i][j] = a[i][j] - a[i][c]*a[f][j]; }  
        for ( j = 0; j < n; j++ )  
        { b[i][j] = b[i][j] - a[i][c]*b[f][j]; }  
        a[i][c] = 0; } }  
// Impresión de la matriz inversa.  
printf("\n\n\tMatriz inversa de la dada");  
for ( i = 0; i < n; i++ )  
{ printf("\n");  
    for ( j = 0; j < n; j++ )  
    { printf("%10f", b[i][j]); } } }  
printf("\n\n");  
}  
void cls(void)  
{  
    printf ("\033[2J");      /* Limpia la pantalla */  
    printf ("\033[0;0f");     /* Mueve el cursor en la esquina superior izquierda */  
}
```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO MATRICES EN LENGUAJE C

/* **Matr15:** Escribir un programa en lenguaje C que:

Permita imprimir estadísticas de ventas de una empresa que comercializa un sólo producto. El informe debe contener:

- a) El total de ventas de cada día del año;
- b) El total de ventas de cada mes;
- c) El total de ventas del año; y,
- d) El vendedor que más artículos vendió en el año.

Los códigos de los vendedores, que son 20, son números enteros consecutivos.

Para el programa generar al azar la matriz de ventas, sabiendo que ningún vendedor comercializó mas de 50 artículos y que

la matriz de ventas contiene el número de artículos vendidos en el año, por mes, día (25) y código de vendedor. */

```
#define intmax 2147483647
#define venmax 50 /* Número máximo de artículos vendidos. */
#define l 12 /* Número de meses del año.
#define m 25 /* Número de días de ventas del mes.
#define n 20 /* Número de vendedores.
#include <stdio.h> /* Para entrada y salida con scanf() y printf()
#include <stdlib.h> /* Para generar números al azar. La función ran() lo genera de 0 a 32767.
void cls(void);
main()
{
    int i, j, k, codven;
    int ventas[l][m][n], vendia[i][j], venmes[], venano;
    float r;
    cls();
    for ( i = 0; i < l; i++ )
    { for ( j = 0; j < m; j++ )
        { for ( k = 0; k < n; k++ )
            { r = rand();
              r = (r/intmax)*venmax + 0.5;
              ventas[i][j][k] = r; } } }
/* */
// Impresión de la matriz de ventas.
for ( i = 0; i < l; i++ )
{ printf ("\n\t\t\tVentas diarias, por vendedor, del mes%3d ", i + 1 );
  printf ("\n\n    vendedor: ");
  for ( k = 0; k < n; k++ )
  { printf (" %4d", k + 1 );
  printf ("\n");
  for ( j = 0; j < m; j++ )
  { printf ("\n\tdía%3d ", j + 1 );
    for ( k = 0; k < n; k++ )
    { printf (" %4d", ventas[i][j][k] );
    printf ("\n\n");
    while ( getchar() != '\n' ) // '\n' es la tecla return.
    { }
    cls(); }
/* */
// Cálculo del total de ventas de cada día del año.
for ( i = 0; i < l; i++ )
{ for ( j = 0; j < m; j++ )
{ vendia[i][j] = 0;
  for ( k = 0; k < n; k++ )
  { vendia[i][j] += ventas[i][j][k]; } } }
```

FACULTAD D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005
EJERCICIOS DE PROGRAMACIÓN UTILIZANDO MATRICES EN LENGUAJE C

```
// Matr15: (Cont.) Impresión del total de ventas de cada día del año.  
cls();  
printf ("\n\t\t\tTotal de ventas de cada día del año\n\n\t\tmes\t");  
for ( i = 0; i < l; i++ )  
{ printf (" %02d", i + 1 ); }  
printf ( "\n\t\tdía");  
for ( j = 0; j < m; j++ )  
{ printf ( "\n\t\t%02d ", j + 1 );  
    for ( i = 0; i < l; i++ )  
        { printf ( "%6d", vendia[i][j] ); } }  
printf ("\n\n");  
while ( getchar() != '\n' )           // '\n' es la tecla return.  
{ }  
// Cálculo del total de ventas de cada mes.  
for ( i = 0; i < l; i++ )  
{ venmes[i] = 0;  
    for ( j = 0; j < m; j++ )  
        { for ( k = 0; k < n; k++ )  
            { venmes[i] += ventas[i][j][k]; } } }  
// Cálculo del total de ventas del año.  
venano = 0;  
for ( i = 0; i < l; i++ )  
{ for ( j = 0; j < m; j++ )  
    { for ( k = 0; k < n; k++ )  
        { venano += ventas[i][j][k]; } } }  
// Impresión del total de ventas de cada mes.  
cls();  
printf ("\n\n\n\t\t\tTotal de ventas de cada mes y total del año\n\n\t");  
for ( i = 0; i < l; i++ )  
{ printf (" %02d", i + 1 ); }  
printf ( " Año\n\n\t");  
for ( i = 0; i < l; i++ )  
{ printf ( "%8d", venmes[i] ); }  
printf ( "%10d", venano );  
printf ("\n\n");  
}  
void cls(void)  
{  
    printf ("\033[2J");      /* Limpia la pantalla */  
    printf ("\033[0;0f");     /* Mueve el cursor en la esquina superior izquierda */  
}
```