

**FACULTAD D 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\t\tFacultad de Ingeniería - COMPUTACIÓN - Ejercicio VS01\n\n\n");
    printf("\n\t\t\t\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\t\t\t\tEl primer número %d es par", a); }
      else
      { printf("\n\t\t\t\t\tEl primer número %d es impar", a); }
      if ( b%2 == 0 )
      { printf("\n\t\t\t\t\tEl segundo número %d es par", b); }
      else
      { printf("\n\t\t\t\t\tEl segundo número %d es impar", b); }
      if ( c%2 == 0 )
      { printf("\n\t\t\t\t\tEl tercer número %d es par", c); }
      else
      { printf("\n\t\t\t\t\tEl tercer número %d es impar", c); }
      if ( d%2 == 0 )
      { printf("\n\t\t\t\t\tEl cuarto número %d es par\n\n", d); }
      else
      { printf("\n\t\t\t\t\tEl cuarto número %d es impar\n\n", d); } }
    else
    { printf("\n\t\t\t\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\t\t\tFacultad de Ingeniería - COMPUTACIÓN - Ejercicio VS02\n\n\n");
    printf("\n\t\t\t\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\t\t\t\tEl primer número %d es múltiplo de 5", a); }
      else
      { printf("\n\t\t\t\t\tEl primer número %d no es múltiplo de 5", a); }
      if ( b%5 == 0 )
      { printf("\n\t\t\t\t\tEl segundo número %d es múltiplo de 5", b); }
      else
      { printf("\n\t\t\t\t\tEl segundo número %d no es múltiplo de 5", b); }
      if ( c%5 == 0 )
      { printf("\n\t\t\t\t\tEl tercer número %d es múltiplo de 5", c); }
      else
      { printf("\n\t\t\t\t\tEl tercer número %d no es múltiplo de 5", c); }
      if ( d%5 == 0 )
      { printf("\n\t\t\t\t\tEl cuarto número %d es múltiplo de 5\n\n", d); }
      else
      { printf("\n\t\t\t\t\tEl cuarto número %d no es múltiplo de 5\n\n", d); } }
    else
    { printf("\n\t\t\t\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**

/\* **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\t\t\tFacultad de Ingeniería - COMPUTACIÓN - Ejercicio VS03\n\n");
    printf("\n\t\t\t\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\t\t\tLos números %d, %d, %d y %d son términos de una progresión aritmética", a, b, c, d); }
      else
      { printf("\n\t\t\t\tLos números %d, %d, %d y %d no son términos de una progresión aritmética", a, b, c, d); } }
    else
    { printf("\n\t\t\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**

/\* **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\t\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\t\t\tLos números %d, %d, %d y %d son términos de una progresión geométrica", a, b, c, d); }
      else
      { printf("\n\t\t\t\tLos números %d, %d, %d y %d no son términos de una progresión geométrica", a, b, c,
d); } }
    else
    { printf("\n\t\t\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**

/\* **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\t\t\tFacultad de Ingeniería - COMPUTACIÓN - Ejercicio VS05\n\n");
    printf("\n\t\t\t\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\t\t\t\tLos números %d, %d, %d y %d están en secuencia ascendente\n\n", a, b, c, d); }
      else
      { printf("\n\t\t\t\t\tLos números %d, %d, %d y %d no están en secuencia ascendente\n\n", a, b, c, d); } }
    else
    { printf("\n\t\t\t\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**

/\* **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\t\t\tFacultad de Ingeniería - COMPUTACIÓN - Ejercicio VS06\n\n\n");
    printf("\n\n\t\t\t\t\t Digite dos números enteros y positivos ");
    scanf("%d %d", &a, &b);
    while ( a <= 0 || b <= 0 )
    { printf("\n\t\t\t\t\t Los números deben ser positivos, digite nuevamete, ");
      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\n\t\t\t\t\t mcd( %d, %d) = %d", a, b, mcd);
    printf("\n\n\t\t\t\t\t mcm( %d, %d) = %d\n\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 D INGENIERÍA – UNA – PRIMER SEMESTRE – COMPUTACIÓN – 2005**  
**EJERCICIOS DE PROGRAMACIÓN UTILIZANDO VARIABLES SIMPLES EN LENGUAJE C**

```

/* VS07: Escribir un programa en lenguaje C que:
   Permite 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\t\t\tFacultad de Ingeniería - COMPUTACIÓN - Ejercicio VS07\n\n\n");
    printf("\t\t\t\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\t\t\t\t\tEl número %d es primo\n\n", a); }
      else
      { printf("\n\n\t\t\t\t\tEl número %d no es primo\n\n", a); } }
    else
    { printf("\n\n\t\t\t\t\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 D 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 %8.3fi\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 D 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\t\t\t\tDigite un número entero y positivo: ");
    scanf("%d", &n); // Lectura del número natural.
    while ( n <= 0 )
    { printf("\n\t\t\t\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\t\t\t\tLa suma de los pares menores o iguales a %d es %d \n", n, spar);
      printf("\n\t\t\t\t\tLa suma de los impares menores a %d es %d \n\n", n, simp); }
    else
    { spar = (n + 1)*(n - 1)/4;
      simp = (n + 1)*(n + 1)/4;
      printf("\n\t\t\t\t\tLa suma de los pares menores a %d es %d \n", n, spar);
      printf("\n\t\t\t\t\tLa suma de los impares menores o iguales a %d es %d \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\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\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:  $\text{suma}(7\text{sen}(3nx)\text{cos}(nx))$ , 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\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 D 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\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\t\t\t\tCon n = %d y el sumando no tomado = %f se tiene que exp(%f) = %f\n\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\t\tDigite el número de datos a considerar: ");
    scanf ("%d", &n);
    printf ("\n\t\t\t\t\tDigite el primer dato a considerar  : ");
    scanf ("%f", &a);
    men = a;
    may = a;
    for ( i = 1; i < n; i++ )
    { printf ("\t\t\t\t\tDigite el siguiente dato a considerar: ");
      scanf ("%f", &a);
      if ( a < men ) men = a;
      if ( a > may ) may = a; }
    printf("\n\n\t\t\t\t\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 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 %d billetes de %9d ", nbil[i], dbil[i]); } }  
printf ("\n\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\t\tDigite el número de componentes de los vectores: ");
    scanf ("%d", &n);
    while ( n <= 0)
    { printf ("\n\t\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 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\t\t\tEl producto escalar es %f\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\t\tDigite el número de componentes del vector: ");
  scanf ("%d", &n);
  while ( n <= 0)
  { printf ("\n\t\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", &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\n\t\t\t\t\tLa menor de las componentes es v[%d] = %f", men, v[men]);
  printf ("\n\n\t\t\t\t\tLa mayor de las componentes es v[%d] = %f\n\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

/\* **Vect05:** Escribir un programa en lenguaje C que:

Permita insertar, una componente dada, en la posición  $m$  en un vector de  $n$  componentes ( $n \geq m$ ).  
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\t Digite el número de componentes del vector: ");
    scanf ("%d", &n);
    while ( n <= 0)
    { printf ("\n\t\t\t\t\t El 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\t\t Digite la componente %d del vector: ", i);
      scanf ("%f", &v[i]); }
    printf ("\n\t\t\t\t\t Digite 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\t\t Posición 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\t\t\t La 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\t\tDigite el número de componentes del vector: ");
  scanf ("%d", &n);
  while ( n <= 0)
  { printf ("\n\t\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", &v[i]); }
  printf ("\n\t\t\t\t\tDigite 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\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\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

```
datos. */
#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\t Digite cuantos números enteros y positivos considerará ");
    scanf ("%d", &n);
    while ( n <= 1 || n > nmax )
    { printf ("\n\t\t El número de datos debe ser mayor que uno y menor que %d, digite nuevamete, ", nmax);
      scanf ("%d", &n); }
    printf ("\n");
    for (i = 0; i < n; i++)
    { printf ("\t\t\t\t Digite nro( %d ) = ", i);
      scanf ("%d", &nros[i]);
      while (nros[i] <= 0 )
      { printf ("El número debe ser positivo, digite nuevamete, ");
        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  $y = 3x^2 - 2x + 4$ , 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\t\tDigite el número de componentes del vector: ");
    scanf ("%d", &n);
    while ( n <= 0 )
    { printf ("\n\t\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", &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 %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 DE 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\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\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**

/\* **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\tDigite el número de componentes del vector: ");
    scanf ("%d", &n);
    while ( n <= 0 )
    { printf ("\n\t\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", &v[i]); }
    printf ("\n\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\tEl elemento dado ocupa la posición %d en el vector", i); }
    else
    { printf ("\n\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 MATRICES EN LENGUAJE C**

```
/* Matr02: Escribir un programa en lenguaje C que:  
  Lea un número n y lo acepte sólo si es entero y positivo, en caso contrario, solicite nuevamente el  
  número n, hasta  
  obtener uno entero y positivo, y los nxn elementos de la matriz cuadrada mat. Imprima luego el  
  número n, los  
  elementos de la matriz mat y tres números r, s y t, obtenidos, respectivamente, como la suma de los  
  elementos de  
  mat situados sobre la diagonal principal, la suma de de los elementos de mat situados debajo de la  
  diagonal principal  
  y la suma de los elementos de mat situados en la diagonal principal. */  
#define nmax 50  
#include <stdio.h>  
void cls(void);  
main()  
{  
  int i, j, n;  
  float r, s, t, mat[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");  
  r = 0;  
  s = 0;  
  t = 0;  
  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]);  
      if ( i < j ) r += mat[i][j];  
      if ( i > j ) s += mat[i][j];  
      if ( i == j ) t += mat[i][j]; } }  
  printf ("\n\t\t\tOrden de la matriz %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\n\t\t\tSuma de los elementos situados sobre la diagonal principal : %10f", r);  
  printf ("\n\t\t\tSuma de los elementos situados debajo de la diagonal principal: %10f", s);  
  printf ("\n\t\t\tSuma de los elementos situados en la diagonal principal : %10f\n\n", t);  
}  
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**

```

/* Matr06: Escribir un programa en lenguaje C que:
   Permita imprimir la traspuesta de una matriz dada de orden mxn. */
#define nmax 50
#include <stdio.h>
void cls(void);
main()
{
    int i, j, m, n;
    float mat[nmax][nmax], matt[nmax][nmax];
    cls();
    printf ("\n\t\t\t\t\tDigite el orden de la matriz (m n): ");
    scanf ("%d %d", &m, &n);
    while ( n <= 0 || n > nmax || m <= 0 || m > nmax )
    { printf ("\n\t\t\tLos números de filas y columnas deben ser enteros, positivos y no mayores que %d,
digite nuevamente: ", nmax);
      scanf ("%d %d", &m, &n); }
    printf ("\n");
    for ( i = 0; i < m; 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]);
        matt[j][i] = mat[i][j]; } }
    cls();
    printf ("\n\n\t\t\tMatriz dada, de orden %3d x%3d", m, n);
    for ( i = 0; i < m; 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\tMatriz traspuesta de la dada, de orden %3d x%3d", n, m);
    for ( i = 0; i < n; i++ )
    { if ( i == 0 )
      { printf ("\n{ "); }
      else
      { printf ("\n {"); }
      for ( j = 0; j < m; j++ )
      { if ( j < m - 1 )
        { printf ("%10f, ", matt[i][j]); }
        else
        { printf ("%10f", matt[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**

```
/* Matr08: Escribir un programa en lenguaje C que permita sumar dos matrices de orden mxn. */
#define nmax 50
#include <stdio.h>
void cls(void);
main()
{
    int i, j, m, n;
    float mata[nmax][nmax], matb[nmax][nmax], mats[nmax][nmax];
    cls();
    printf ("\n\t\t\t\t\tDigite el orden de la matriz (m n): ");
    scanf ("%d %d", &m, &n);
    while ( m <= 0 || m > nmax || n <= 0 || n > nmax )
    { printf ("\n\t\tLos números de filas y columnas deben ser enteros, positivos y no mayores que %d, digite nuevamente: ",
nmax);
        scanf ("%d %d", &m, &n); }
    printf ("\n\t\tDigite las componentes de la primera matriz sumando\n");
    for ( i = 0; i < m; i++ )
    { for ( j = 0; j < n; j++ )
        { printf ("\t\t\t\t\tDigite la componente mata[%d][%d]: ", i, j);
          scanf ("%f", &mata[i][j]); } }
    printf ("\n\t\tDigite las componentes de la segunda matriz sumando\n");
    for ( i = 0; i < m; i++ )
    { for ( j = 0; j < n; j++ )
        { printf ("\t\t\t\t\tDigite la componente matb[%d][%d]: ", i, j);
          scanf ("%f", &matb[i][j]); } }
    for ( i = 0; i < m; i++ )
    { for ( j = 0; j < n; j++ )
        { mats[i][j] = mata[i][j] + matb[i][j]; } }
    cls();
    printf ("\n\n\t\tPrimera matriz dada, de orden %3d x%3d", m, n);
    for ( i = 0; i < m; i++ )
    { if ( i == 0 )
        { printf ("\n{ "); }
        else
        { printf ("\n {"); }
        for ( j = 0; j < n; j++ )
        { if ( j < n - 1 )
            { printf ("%10f, ", mata[i][j]); }
            else
            { printf ("%10f}", mata[i][j]); } } }
    printf (" }\n\n\t\tSegunda matriz dada, de orden %3d x%3d", m, n);
    for ( i = 0; i < m; i++ )
    { if ( i == 0 )
        { printf ("\n{ "); }
        else
        { printf ("\n {"); }
        for ( j = 0; j < n; j++ )
        { if ( j < n - 1 )
            { printf ("%10f, ", matb[i][j]); }
            else
            { printf ("%10f}", matb[i][j]); } } }
    printf (" }\n\n\t\tMatriz suma de las matrices dadas, de orden %3d x%3d", m, n);
    for ( i = 0; i < m; i++ )
    { if ( i == 0 )
        { printf ("\n{ "); }
        else
        { printf ("\n {"); }
        for ( j = 0; j < n; j++ )
        { if ( j < n - 1 )
            { printf ("%10f, ", mats[i][j]); }
            else
            { printf ("%10f}", mats[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**

```
// Matr12: (Cont.) Impresión del sistema de ecuaciones dado.
cls();
printf("\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\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++ )
      { a[f][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\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\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

```
// Matr14: (Cont.) Impresión de la matriz dada.
cls();
printf ("\n\n\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\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\t\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[l][m], venmes[l], 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\t\t\tvendedor: ");
  for ( k = 0; k < n; k++ )
  { printf ( "%4d", k + 1 ); }
  printf ("\n");
  for ( j = 0; j < m; j++ )
  { printf ("\n\t\t\t día%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\t\t\tTotal de ventas de cada día del año\n\n\t\t\t\t\t mes\t ");
for ( i = 0; i < l; i++ )
{ printf ( " %02d", i + 1 ); }
printf ( "\n\t\t\t\t\t día");
for ( j = 0; j < m; j++ )
{ printf ( "\n\t\t\t\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\t\t\tTotal de ventas de cada mes y total del año\n\n\n\t\t\t\t\t ");
for ( i = 0; i < l; i++ )
{ printf ( " %02d", i + 1 ); }
printf ( " Año\n\n\t\t\t\t\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 */
}
}
```