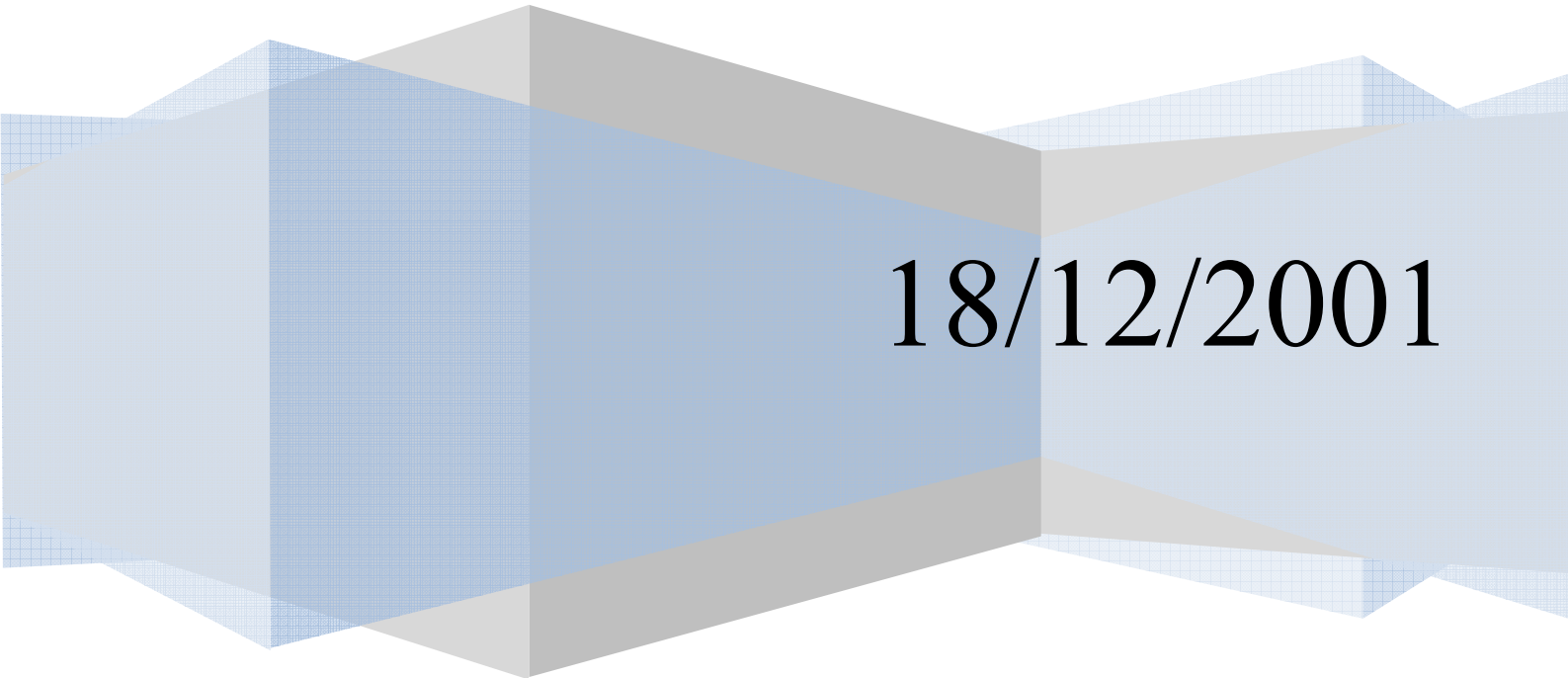


:: [www.vcgs.net](http://www.vcgs.net) :: Víctor Campuzano Gallego

# Algoritmos en pseudocódigo y su traducción a lenguaje C

Ciclo Formación de Grado Superior de  
Administración de Sistemas Informáticos

*Módulo de Fundamentos de Programación*



18/12/2001

**RELACIÓN N°1**

**1.- Algoritmo que toma como dato de entrada un número que corresponde a la longitud de un radio y nos calcula y escribe la longitud de la circunferencia, el área del círculo y el volumen de la esfera que se corresponden con dicho radio.**

**PROGRAMA:**

Area\_Volumen\_Longitud

**ENTORNO:**

PI ← Constante

Radio ← Variable Numérica real

Volumen ← Variable Numérica real

Area ← Variable Numérica real

Longitud ← Variable Numérica real

**ALGORITMO:**

PI ← 3.141592

Pedir Radio

Longitud ←  $2 * PI * Radio$

Area ←  $PI * Radio^2$

Volumen ←  $(4 * PI * Radio^3) / 3$

Escribir "La Longitud de la circunferencia es:", Longitud

Escribir "El Area del circulo es:", Area

Escribir "El Volumen de la esfera es:", Volumen

**FIN ALGORITMO**

-----  
/\* Relación 1. Ej - 1 \*/

#include <stdio.h>

#define PI 3.141592

main ()

{

int Radio;

float Volumen, Area, Longitud;

printf ("Escriba el valor del Radio: ");

scanf ("%d", &Radio);

Logitud =  $2 * PI * Radio$ ;

Area =  $PI * Radio * Radio$ ;

Volumen =  $(4 * PI * Radio * Radio * Radio) / 3$ ;

printf ("Longitud %.6f\n", Longitud);

printf ("Area %.6f\n", Area);

printf ("Volumen %.6f\n", Volumen);

}

**2.- Algoritmo que lee 2 números, calculando y escribiendo el valor de su suma, resta, producto y división.**

**PROGRAMA:**

Lo\_Hace\_Todo

**ENTORNO:**

Numero\_1 ← Numérico entero  
Numero\_2 ← Numérico entero  
Suma ← Numérico entero  
Resta ← Numérico entero  
Producto ← Numérico entero  
Division ← Numérico entero

**ALGORITMO:**

Leer Numero\_1  
Leer Numero\_2  
Suma ← Numero\_1 + Numero\_2  
Resta ← Numero\_1 - Numero\_2  
Producto ← Numero\_1 \* Numero\_2  
Division ← Numero\_1 / Numero\_2  
Escribir "Suma =", Suma  
Escribir "Resta =", Resta  
Escribir "Producto =", Producto  
Escribir "División =", Division

**FIN ALGORITMO**

```
-----  
/* Relación 1. Ej - 2 */  
#include <stdio.h>  
main ()  
{  
    int X, Y, Suma, Resta;  
    float Producto, División;  
    printf ("Escriba Numero A: ");  
    scanf ("%d", &X);  
    printf ("Escriba Numero B: ");  
    scanf ("%d", &Y);  
    Suma = X + Y;  
    Resta = X - Y;  
    Producto = X * Y;  
    División = X / Y;  
    printf ("La suma es: %d\n", Suma);  
    printf ("La resta es: %d\n", Resta);  
    printf ("El producto es: %.0f\n", Producto);  
    printf ("La división es: %.2f\n", División);  
}
```

**3.- Algoritmo que lee 2 números y nos dice cuál es el mayor o si son iguales.**

**PROGRAMA:**

Mayor\_O\_Igual

**ENTORNO:**

Numero\_1 ← Numérico entero  
Numero\_2 ← Numérico entero

**ALGORITMO:**

```
Leer Numero_1
Leer Numero_2
Si Numero_1 > Numero_2 Entonces
    Escribir Numero_1, "mayor que", Numero_2
Si no
    Si Numero_1 < Numero_2 Entonces
        Escribir Numero_1, "menor que", Numero_2
    Si no
        Escribir Numero_1, "Igual que", Numero_2
    Fin Si
Fin Si
```

**Fin Si**  
**FIN ALGORITMO**

```
-----
/* Relacion 1. Ej - 3 */
#include <stdio.h>
main ()
{
    int Numero_1, Numero_2;
    printf ("Dame el primer Numero: ");
    scanf ("%d", &Numero_1);
    printf ("Dame el segundo Numero: ");
    scanf ("%d", &Numero_2);
    if (Numero_1 > Numero_2)
    {
        printf ("%d Mayor que %d", Numero_1, Numero_2);
    }
    else
    {
        if (Numero_1 < Numero_2)
        {
            printf ("%d Menor que %d", Numero_1, Numero_2);
        }
        else
        {
            printf ("%d Igual que %d", Numero_1, Numero_2);
        }
    }
}
```

**4.- Algoritmo que lee 3 números distintos y nos dice cual de ellos es el mayor.**

**PROGRAMA:**

Cual\_es\_Mayor

**ENTORNO:**

Numero\_1 ← Numerico entero  
Numero\_2 ← Numerico entero  
Numero\_3 ← Numerico entero

**ALGORITMO:**

```
Leer Numero_1
Leer Numero_2
Leer Numero_3
Si Numero_1 > Numero_2 Entonces
    Si Numero_1 > Numero_3 Entonces
        Escribir Numero_1, "es el mayor"
    Si no
        Escribir Numero_3, "es el mayor"
    Fin Si
Si no
    Si Numero_2 > Numero_3 Entonces
        Escribir Numero_2, "es el mayor"
    Si no
        Escribir Numero_3, "es el mayor"
    Fin si
Fin si
FIN ALGORITMO
```

```
-----
/* Relacion 1. Ej - 4. */
#include <stdio.h>
main()
{
    int A,B,C;
    printf ("Escriba número A: ");
    scanf ("%d", &A);
    printf ("Escriba número B: ");
    scanf ("%d", &B);
    printf ("Escriba número C: ");
    scanf ("%d", &C);
    if (A>B)
    {
        if (A>C)
        {
            printf ("%d Es el mayor..", A);
        }
        else
        {
            printf ("%d Es el mayor..", C);
        }
    }
    else
    {
        if (B>C)
        {
            printf ("%d Es el mayor..", B);
        }
    }
}
```

```
        else
        {
            printf ("%d Es el mayor..", C);
        }
    }
}
```

**5.- Algoritmo que lee una temperatura en la escala centígrada y nos calcula y escribe su valor en las escalas Reamur, Fahrenheit, Kelvin.**

**PROGRAMA:**

Multi\_Temperatura

**ENTORNO:**

Tcent ← Numéricas reales

Tream ← Numéricas reales

Tfaren ← Numéricas reales

Tkelv ← Numéricas reales

**ALGORITMO:**

Leer Tcent

Tream ← (80 \* Tcent) / 100

Tfaren ← (1.8 \* Tcent) + 32

Tkelv ← (Tcent - 273)

Escribir Tcent, " °Centígrados =", Tfaren, " ° Fahrenheit"

Escribir Tcent, " °Centígrados =", Tream, " ° Reamur"

Escribir Tcent, " °Centígrados =", Tkelv, " ° Kelvin"

**FIN ALGORITMO**

-----  
/\* Relación 1. Ej - 5 \*/

#include <stdio.h>

main ()

```
{
    float Tcent, Tream, Tfaren, Tkelv;
    printf ("Escriba el valor en grados centigrados: ");
    scanf ("%f", &Tcent);
    Tream = 0.8 * Tcent;
    Tfaren = (1.8 * Tcent) + 32;
    Tkelv = Tcent - 273;
    printf ("%f Centigrados => %f Farenheit\n", Tcent, Tfaren);
    printf ("%f Centigrados => %f Reamur\n", Tcent, Tream);
    printf ("%f Centigrados => %f Kelvin\n", Tcent, Tkelv);
}
```

**6.- Algoritmo que lee tres números cualesquiera y los escribe ordenados de forma ascendente (descendente).**

**PROGRAMA:**

Ordena\_Tres

**ENTORNO:**

X ← Numérica entera  
Y ← Numérica entera  
Z ← Numérica entera  
Auxi ← Numérica entera  
Ascendente ← Booleana (Cierto – Falso)

**ALGORITMO:**

Leer X  
Leer Y  
Leer Z  
Leer Ascendente  
Si Z > X Entonces  
    Auxi ← X  
    X ← Z  
    Z ← Auxi  
Fin Si  
Si Z > Y Entonces  
    Auxi ← Y  
    Y ← Z  
    Z ← Auxi  
Fin Si  
Si Y > X Entonces  
    Auxi ← X  
    X ← Y  
    Y ← Auxi  
Fin Si  
Si Ascendente = Cierto Entonces  
    Escribir X, "mayor o igual que", Y, "mayor o igual que", Z  
Si no  
    Escribir Z, "menor o igual que", Y, "menor o igual que", X  
Fin Si  
**FIN ALGORITMO**

-----

```
/* Relacion 1. Ej - 6 */  
#include <stdio.h>  
main ()  
{  
    int X, Y, Z, Auxi;  
    printf ("Escriba un numero entero: ");  
    scanf ("%d", &X);  
    printf ("Escriba otro numero entero: ");  
    scanf ("%d", &Y);  
    printf ("Escriba el ultimuo numero: ");  
    scanf ("%d", &Z);  
    if (Z>X)  
    { Auxi = X;  
      X = Z;  
      Z = Auxi;
```

```
    }  
    if (Z>Y)  
    {  
        Aux1 = Y;  
        Y = Z;  
        Z = Aux1;  
    }  
    if (Y>X)  
    { Aux1 = X;  
      X = Y;  
      Y = Aux1;  
    }  
    printf ("%d - %d - %d", X, Y, Z);  
}
```

**7.- Algoritmo que lee una calificación numérica entre 0 y 10 y la transforma en calificación alfabética, escribiendo el resultado.**

**PROGRAMA:**

Califica

**ENTORNO:**

Calif ← Numérico entero

**ALGORITMO:**

Leer Calif

Mientras Calif < 0 o Calif > 10 Hacer

    Escribir "Dato erróneo, escríbalo otra vez"

    Leer Calif

Fin Mientras

Si Calif < 5 Entonces

    Escribir "INSUFICIENTE"

Si no

    Si Calif >= 5 Y Calif < 6 Entonces

        Escribir "SUFICIENTE"

    Si no

        Si Calif >= 6 Y Calif < 7 Entonces

            Escribir "BIEN"

        Si no

            Si Calif >= 7 Y Calif < 9 Entonces

                Escribir "NOTABLE"

            Si no

                Escribir "SOBRESALIENTE"

            Fin Si

        Fin Si

    Fin Si

Fin Si

**FIN ALGORITMO**

-----



```
/* Relacion 1. Ej - 7 */
#include <stdio.h>
main ()
{
    int Nota;
    printf ("Escriba la nota: ");
    scanf ("%d", &Nota);
    switch (Nota)
    {
        case 1: case 2: case 3: case 4:
        {
            printf ("Insuficiente\n");
        }
        case 5:
        {
            printf ("Aprobado\n");
        }
        case 6:
        {
            printf ("Bien\n");
        }
        case 7: case 8:
        {
            printf ("Notable\n");
        }
        case 9: case 10:
        {
            printf ("Sobresaliente\n");
        }
        case else:
        {
            printf ("Nota errónea, fin del programa\n");
        }
    }
}
```

**8.- Algoritmo que lee tres números cualquiera y nos indica todas sus relaciones de igualdad.**

**PROGRAMA:**

Relaciones\_Igualdad

**ENTORNO:**

X ← Numero Real

Y ← Numero Real

Z ← Numero Real

**ALGORITMO:**

Leer X

Leer Y

```
Leer Z
Si X > Y Entonces
    Escribir X, "Mayor Que", Y
    Escribir Y, "Menor Que", X
Si no
    Si X = Y Entonces
        Escribir X, "Es igual a", Y
    Si no
        Escribir Y, "Mayor que", X
        Escribir X, "Menor que", Y
    Fin Si
Fin si
Si X > Z Entonces
    Escribir X, "Mayor Que", Z
    Escribir Z, "Menor Que", X
Si no
    Si X = Z Entonces
        Escribir X, "Es igual a", Z
    Si no
        Escribir Z, "Mayor que", X
        Escribir X, "Menor que", Z
    Fin Si
Fin si
Si Y > Z Entonces
    Escribir Y, "Mayor Que", Z
    Escribir Z, "Menor Que", Y
Si no
    Si Y = Z Entonces
        Escribir Y, "Es igual a", Z
    Si no
        Escribir Z, "Mayor que", Y
        Escribir Y, "Menor que", Z
    Fin Si
Fin si
FIN ALGORITMO
```

```
-----
/* Relacion 1. Ej - 8 */
#include <stdio.h>
main ()
{
    int X, Y, Z;
    printf ("Escriba numero A: ");
    scanf ("%d", &X);
    printf ("Escriba numero B: ");
    scanf ("%d", &Y);
    printf ("Escriba numero C: ");
    scanf ("%d", &Z);
```

```
if (X>Y)
{
    printf ("%d Mayor que %d\n", X, Y);
    printf ("%d Menor que %d\n", Y, X);
}
else
{
    if (X == Y)
    {
        printf ("%d Igual que %d\n", X, Y);
    }
    else
    {
        printf ("%d Mayor que %d\n", Y, X);
        printf ("%d Menor que %d\n", X, Y);
    }
}
if (X>Z)
{
    printf ("%d Mayor que %d\n", X, Z);
    printf ("%d Menor que %d\n", Z, X);
}
else
{
    if (X == Z)
    {
        printf ("%d Igual que %d\n", X, Z);
    }
    else
    {
        printf ("%d Mayor que %d\n", Z, X);
        printf ("%d Menor que %d\n", X, Z);
    }
}
if (Y>Z)
{
    printf ("%d Mayor que %d\n", Y, Z);
    printf ("%d Menor que %d\n", Z, Y);
}
else
{
    if (Y == Z)
    {
        printf ("%d Igual que %d\n", Y, Z);
    }
    else
    {
        printf ("%d Mayor que %d\n", Z, Y);
    }
}
```

```
        printf ("%d Menor que %d\n", Y, Z);
    }
}
}
```

9.- En un determinado comercio, se realiza un descuento dependiendo del precio de cada producto. Si el precio es inferior a 1000 pts, no se hace descuento; si es mayor o igual a 1000 ptas y menor que 10000 ptas, se hace un 5% de descuento, y si es mayor o igual a 10000 ptas, se hace un 10% de descuento.

Algoritmo que lee el precio de un producto y nos calcula y escribe su precio final.

**PROGRAMA:**

Calcula\_Dto

**ENTORNO:**

Total ← Numerico entero

Importe ← Numerico entero

Dto ← Numerico entero

**ALGORITMO:**

Leer Importe

Si Importe < 1000 Entonces

Dto ← 0

Si no

Si Importe < 10000 Entonces

Dto ← 5

Si no

Dto ← 10

Fin Si

Fin Si

Total ← Importe - ((Importe \* Dto) / 100)

Escribir "El total a pagar es: ", Total

**FIN ALGORITMO**

-----  
/\* Relacion 1. Ej - 9 \*/

#include <stdio.h>

main ()

{

float Impor, Dto=0;

float Total;

printf ("Escriba el importe: ");

scanf ("%f", &Impor);

if (Impor < 1000)

Dto = 0;

else

if (Impor < 10000)

Dto = 5;

else

```
        Dto = 10;
    Total = Impor - ((Impor * Dto) / 100);
    printf ("El total a pagar es: %.0f", Total);
}
```

**10.- Algoritmo que lee 2 números enteros, positivos y distintos y nos dice si el mayor es múltiplo del menor o si el menor es divisor del mayor.**

**PROGRAMA:**

¿Divisible?

**ENTORNO:**

Numero\_Mayor ← Numero entero

Numero\_Menor ← Numero entero

Auxiliar ← Numero entero

**ALGORITMO:**

Leer Numero\_Mayor

Leer Numero\_Menor

Mientras Numero\_Menor <= 0 Hacer

    Escribir "El divisor es erróneo, escriba otro"

    Leer Numero\_Menor

Fin Mientras

Auxiliar ← 1

Mientras Auxiliar < Numero\_Mayor Hacer

    Auxiliar ← Auxiliar + Numero\_Menor

Fin Mientras

Si Auxiliar = Numero\_Mayor Entonces

    Escribir Numero\_Mayor, "es múltiplo de", Numero\_Menor

Si no

    Escribir Numero\_Mayor, "no es múltiplo de", Numero\_Menor

Fin Si

**FIN ALGORITMO**

-----  
/\* Relacion 1. Ej - 10 \*/

#include <stdio.h>

main ()

{

    int NuMayor, NuMenor, Aux=0;

    printf ("Escriba el numero mayor: ");

    scanf ("%d", &NuMayor);

    printf ("Escriba el numero menor: ");

    scanf ("%d", &NuMenor);

    while (NuMenor <= 0)

    {

        printf ("El divisor es erróneo, escriba otro: ");

        scanf ("%d", &NuMenor);

    }

    while (Aux < NuMayor)

    {

```
        Auxi = Auxi + NuMenor;
    }
    if (Auxi == NuMayor)
    {
        printf ("%d es multiplo de %d\n", NuMayor, NuMenor);
    }
    else
    {
        printf ("%d no es múltiplo de %d\n", NuMayor, NuMenor);
    }
}
```

**RELACIÓN N° 2.**

**1.- Algoritmo que lee 100 números y cuenta cuántos de ellos son positivos (mayores que 0).**

PROGRAMA:

Cuenpos.psc

ENTORNO:

ConSum, ConInt, Numero -> Numeros enteros

ALGORITMO:

ConSum <- 0

ConInt <- 0

Mientras (ConInt < 100) Hacer

Leer Numero

Si (Numero > 0) Entonces

ConSum <- ConSum + 1

Fin Si

ConInt <- ConInt + 1

Fin Mientras

Escribir "Usted ha introducido ", ConSum, "numeros positivos"

FIN ALGORITMO

---

/\* Cuenta Positivos. Rel 2. Ej - 1 \*/

#include <stdio.h>

main ()

{

int ConSum=0, ConInt=0, Numero;

while (ConInt < 100)

{

printf ("Introduzca un numero, lleva %d\n", ConInt);

scanf ("%d", &Numero);

if (Numero > 0)

{

ConSum = ConSum + 1;

}

ConInt = ConInt + 1;

}

printf ("Usted ha introducido %d Numeros positivos\n", ConSum);

}

**2.- Algoritmo que calcula y escribe la suma y el producto de los 10 primeros números naturales.**

PROGRAMA:

10nat.psc

ENTORNO:

Numero, Suma, Producto -> Numeros enteros

ALGORITMO:

Numero = 1

```
Suma = 0
Producto = 1
Mientras (Numero <= 10) Hacer
    Suma = Suma + Numero
    Producto = Producto * Numero
    Numero = Numero + 1
Fin Mientras
Escribir "La suma total es => ", Suma
Escribir "El producto total es => ", Producto
FIN ALGORITMO
```

---

```
/* 10 primeros numeros naturales. Rel 2. Ej - 2 */
#include <stdio.h>
main ()
{
    int Numero=1, Suma=0, Producto=1;
    while (Numero <= 10)
    {
        Suma = Suma + Numero;
        Producto = Producto * Numero;
        Numero++;
    }
    printf("La suma total es => %d\n", Suma);
    printf("El producto total es => %d\n", Producto);
}
```

**3.- Algoritmo que lee una secuencia de notas (con valores que van de 0 a 10), que termina con el valor -1 y que nos dice si hubo algún 10.**

PROGRAMA:

hay10.psc

ENTORNO:

Nota -> Numero entero

Diez -> Caracter

ALGORITMO:

Leer Nota

Diez = 'n'

Mientras (Nota != -1) Hacer

    Si (Nota < -1 o Nota > 10) Entonces

        Escribir "Dato erróneo"

    Sino

        Si (Nota = 10) Entonces

            Diez = 's'

        Fin Si

    Leer Nota

Fin Mientras

Si (Diez = 's') Entonces

    Escribir "Si ha habido algun 10"

Sino

    Escribir "No, no ha habido ningún 10"



```
Fin Si
FIN ALGORITMO
/* Relacion 2. Ej - 3 */
#include <stdio.h>
main ()
{
    int Nota;
    char Diez='n';
    printf ("Escriba la nota: ");
    scanf ("%d", &Nota);
    while (Nota != -1)
    {
        if (Nota < -1 || Nota>10)
        {
            printf ("Dato erroneo\n");
        }
        else
        {
            if (Nota == 10)
            {
                Diez = 's';
            }
        }
        printf ("Escriba otra nota: ");
        scanf ("%d", &Nota);
    }
    if (Diez == 's')
    {
        printf ("Si, ha habido alg\xn 10\n");
    }
    else
    {
        printf ("No, no ha habido ningun 10\n");
    }
}
```

**4.- Algoritmo que lee una secuencia de 199 números y nos dice cuántos de ellos son positivos y cuántos son negativos.**

PROGRAMA:

posyneg.psc

ENTORNO:

cont, CoPos, CoNeg, Numero -> Numeros enteros

ALGORITMO:

CoPos = 0

CoNeg = 0

Para (cont=1 hasta cont=100 con incremento de 1) Hacer

Leer Numero

Si (Numero >= 0) Entonces

```
        CoPos = CoPos + 1
    Sino
        CoNeg = CoNeg + 1
    Fin Si
Fin Para
Escribir "Han sido ", CoPos, "numeros Positivos"
Escribir "Han sido ", CoNeg, "numeros Negativos"
FIN ALGORITMO
```

---

```
/* Relacion 2. Ej - 4 */
#include <stdio.h>
main ()
{
    int cont;
    int CoPos=0, CoNeg=0,Numero;
    for (cont=1;cont<=100;cont++)
    {
        printf ("Escriba el numero que quiera: ");
        scanf ("%d", &Numero);
        if (Numero >= 0)
        {
            CoPos= CoPos+1;
        }
        else
        {
            CoNeg= CoNeg+1;
        }
    }
    printf ("Usted ha introducido %d numeros Positivos\n", CoPos);
    printf ("Usted ha introducido %d numeros Negativos\n", CoNeg);
}
```

**5.- Algoritmo que lee una secuencia de números no nulos, terminada con la introduccion del valor 0, y obtiene e imprime el mayor.**

PROGRAMA:

    elmayor.psc

ENTORNO:

    Numero, Mayor -> Numeros enteros

ALGORITMO:

Leer Numero

Mayor = Numero

Mientras (Numero != 0) Hacer

    Si (Numero > Mayor) Entonces

        Mayor = Numero

    Fin Si

    Leer Numero

Fin Mientras

Escribir "El mayor numero introducido es => ", Mayor

FIN ALGORITMO

---

```
/* Relacion 2. Ej - 5. */
#include <stdio.h>
main ()
{
    int Numero, Mayor;
    printf ("Escriba un número: ");
    scanf ("%d", &Numero);
    Mayor = Numero;
    while (Numero != 0)
    {
        if (Numero > Mayor)
        {
            Mayor = Numero;
        }
        printf ("Siga escribiendo numeros...: ");
        scanf ("%d", &Numero);
    }
    printf ("El mayor numero introducido es => %d\n", Mayor);
}
```

**6.- Algoritmo que lee un número entero positivo y calcula su factorial.**

PROGRAMA:

facto.psc

ENTORNO:

Numero, Contador -> Numeros enteros

Resultado -> Numero real (Coma flotante)

ALGORITMO:

Leer Numero

Contador = Numero - 1

Resultado = Numero

Mientras (Numero <= 0) Hacer

    Si (Numero = 0) Entonces

        Escribir "Factorial de 0 vale 1, por definición"

    Si no

        Escribir "No existe factorial de un numero negativo"

    Fin Si

    Leer Numero

    Contador = Numero - 1

    Resultado = Numero

Fin Mientras

Mientras (Contador > 1) Hacer

    Resultado = Resultado \* Contador

    Contador = Contador + 1

Fin Mientras

Escribir "Factorial de ", Numero, "vale =>", Resultado

FIN ALGORITMO

---

/\* Relacion 2. Ej - 6 \*/

```
#include <stdio.h>
main ()
{
    int Numero, Contador;
    float Resultado;
    system ("cls");
    printf ("Escriba el numero: ");
    scanf ("%d", &Numero);
    Contador = Numero - 1;
    Resultado = Numero;
    while (Numero <= 0)
    {
        if (Numero == 0)
        {
            printf ("Factorial de 0 vale 1, por definici n\n");
        }
        else
        {
            printf ("No existe el factorial de un numero negativo\n");
        }
        printf ("Escribalo otra vez: ");
        scanf ("%d", &Numero);
        Contador = Numero - 1;
        Resultado = Numero;
    }
    while (Contador > 1)
    {
        Resultado = Resultado * Contador;
        Contador = Contador - 1;
    }
    printf ("Factorial de %d es %.0f", Numero, Resultado);
}
```

**7.- Algoritmo que obtiene el producto de dos n meros enteros positivos mediante sumas sucesivas.**

PROGRAMA:

multi.psc

ENTORNO:

Numero\_1, Numero\_2, Resultado -> Numeros enteros (long int)

Contador -> Numero entero

ALGORITMO:

Leer Numero\_1

Leer Numero\_2

Resultado = Numero\_1

Contador = Numero\_2

Mientras (Contador > 1) Hacer

    Resultado = Resultado + Numero\_1

    Contador = Contador + 1

Fin Mientras  
Escribir "El producto es => ", Resultado  
FIN ALGORITMO

---

```
/* Relacion 2 .- Ej. 7. Multiplica a partir de sumas sucesivas */
#include <stdio.h>
main ()
{
    long int Numero_1, Numero_2, Resultado;
    int Contador;
    printf ("Escriba un numero: ");
    scanf ("%D", &Numero_1);
    printf ("Escriba otro numero: ");
    scanf ("%d", &Numero_2);
    Resultado = Numero_1;
    Contador = Numero_2;
    while (Contador > 1)
        {Resultado = Resultado + Numero_1;
        Contador = Contador - 1;}
    printf ("El producto es, %d", Resultado);
}
```

**8.- Algoritmo que obtiene el cociente y el resto de dos números enteros positivos mediante restas sucesivas.**

PROGRAMA:

divide.psc

ENTORNO:

Dividendo, Divisor, Cociente, Resto -> Numeros enteros

ALGORITMO:

Leer Dividendo

Leer Divisor

Si (Divisor=0) Entonces

Escribir "Cualquier numero dividido por 0 da infinito"

Sino

Cociente = 0

Resto = Dividendo

Mientras (Resto >= Divisor) Hacer

Resto = Resto - Divisor

Cociente = Cociente + 1

Fin Mientras

Escribir "El cociente es => ", Cociente

Escribir "El resto es => ", Resto

Fin Si

FIN ALGORITMO

---

```
/* Relacion 2 .- Ej. 8. Divisor a partir de restas sucesivas */
#include <stdio.h>
main ()
{
    int Dividendo, Divisor, Cociente, Resto;
```

```
printf ("Escriba el Dividendo: ");
scanf ("%d", &Dividendo);
printf ("Escriba el divisor: ");
scanf ("%d", &Divisor);
if (Divisor == 0)
{
    printf ("Cualquier numero dividido por 0 da infinito \n");
}
else
{
    Cociente = 0;
    Resto = Dividendo;
    while (Resto >= Divisor)
    {
        Resto = Resto - Divisor;
        Cociente = Cociente + 1;
    }
    printf ("El cociente es: %d\n", Cociente);
    printf ("El resto es: %d\n", Resto);
}
}
```

**9.- Algoritmo que lee un numero y obtiene su tabla de multiplicar.**

PROGRAMA:

multitab.psc

ENTORNO:

Numero, cont, resul -> Numeros enteros

ALGORITMO:

Leer Numero

Para (cont=0 hasta cont =10 con incremento de 1) Hacer

    resul = Numero \* cont

    Escribir Numero, "por" cont, "=", resul

Fin Para

FIN ALGORITMO

---

/\* Relacion 2. Ej - 9 \*/

#include <stdio.h>

main ()

```
{
    int Numero, cont,resul;
    printf ("De que numero desea ver la tabla de multiplicar? => ");
    scanf ("%d", &Numero);
    for (cont=0; cont<=10; cont++)
    {
        resul = Numero * cont;
        printf ("%d por %d => %d\n", Numero, cont, resul);
    }
}
```

**10.- Algoritmo que lee tres números y los imprime en orden creciente.**

**PROGRAMA:**

Ordena\_Tres

**ENTORNO:**

X ← Numérica entera

Y ← Numérica entera

Z ← Numérica entera

Auxi ← Numérica entera

Ascendente ← Booleana (Cierto – Falso)

**ALGORITMO:**

Leer X

Leer Y

Leer Z

Leer Ascendente

Si Z > X Entonces

Auxi ← X

X ← Z

Z ← Auxi

Fin Si

Si Z > Y Entonces

Auxi ← Y

Y ← Z

Z ← Auxi

Fin Si

Si Y > X Entonces

Auxi ← X

X ← Y

Y ← Auxi

Fin Si

Si Ascendente = Cierto Entonces

Escribir X, "mayor o igual que", Y, "mayor o igual que", Z

Si no

Escribir Z, "menor o igual que", Y, "menor o igual que", X

Fin Si

**FIN ALGORITMO**

---

/\* Relacion 1. Ej - 6 \*/

#include <stdio.h>

main ()

{

int X, Y, Z, Auxi;

printf ("Escriba un numero entero: ");

scanf ("%d", &X);

printf ("Escriba otro numero entero: ");

scanf ("%d", &Y);

printf ("Escriba el ultimuo numero: ");

```
scanf ("%d", &Z);
if (Z>X)
    { Auxi = X;
      X = Z;
      Z = Auxi;
    }
if (Z>Y)
    {
      Auxi = Y;
      Y = Z;
      Z = Auxi;
    }
if (Y>X)
    { Auxi = X;
      X = Y;
      Y = Auxi;
    }
printf ("%d - %d - %d", X, Y, Z);
}
```

**11.- Algoritmo que genera un listado de los N primeros números primos, siendo N el dato de entrada.**

PROGRAMA:

nprim.psc

ENTORNO:

CanPrim, N, ContN, ContRes -> Numeros enteros

ALGORITMO:

N = 1

Escribir N

CanPrim = CanPrim - 1

Mientras (CanPrim > 0) Hacer

ContN = 0

ContRes = 0

Mientras (ContN <= N) Hacer

Si (N%ContN = 0) Entonces

ContRes = ContRes + 1

Fin Si

ContN = ContN + 1

Fin Mientras

Si (ContRes = 2) Entonces

Escribir N

CanPrim = CanPrim + 1

Fin Si

N = N + 1

Fin Mientras

FIN ALGORITMO



```
/* Relacion 2. Ej - 11 */
#include <stdio.h>
main ()
{
    int CanPrim, N, ContN, ContRes;
    printf ("¿Cuantos primos desea incluir en la lista?: ");
    scanf ("%d", &CanPrim);
    N = 1;
    printf ("%d\n", N);
    CanPrim=CanPrim -1;
    while (CanPrim > 0)
    {
        ContN = 1;
        ContRes = 0;
        while (ContN <= N)
        {
            if (N%ContN == 0)
            {
                ContRes = ContRes + 1;
            }
            ContN = ContN + 1;
        }
        if (ContRes == 2)
        {
            printf ("%d\n", N);
            CanPrim = CanPrim -1;
        }
        N = N +1;
    }
}
```

## **12.- Algoritmo que imprime los números perfectos menores de 1000.**

PROGRAMA:

numper.psc

ENTORNO:

N, cont, resul -> Numeros enteros

ALGORITMO

N = 1

Mientras (N < 1000) Hacer

    resul = 0

    cont = 1

    Mientras (cont < N) Hacer

        Si (N%cont = 0) Entonces

            resul = resul + cont

        Fin Si

        cont = cont + 1

    Fin Mientras

```
Si (resul = N) Entonces
    Escribir N
Fin Si
N = N + 1
Fin Mientras
FIN ALGORITMO
```

---

```
/*
Realcion 2. Ej- 12
*/
#include <stdio.h>
main ()
{
    int N=1,cont,resul;
    while (N < 1000)
    {
        resul = 0;
        cont =1;
        while (cont < N)
        {
            if (N%cont == 0)
            {
                resul = resul + cont;
            }
            cont = cont + 1;
        }
        if (resul == N)
        {
            printf ("%d\n", N);
        }
        N = N + 1;
    }
}
```

### **13.- Algoritmo que determina si dos números enteros positivos son amigos.**

PROGRAMA:

primitos.psc

ENTORNO:

N1, N2, cont, suma -> Numeros enteros

ALGORITMO:

cont = 1

suma = 0

Leer N1

Leer N2

Mientras (cont < N1) Hacer

Si (N1%cont = 0) Entonces

suma = suma + cont

Fin Si

```
    cont = cont + 1
Fin Mientras
Si (suma = N2) Entonces
    suma = 0
    cont = 1
    Mientras (cont < N2) Hacer
        Si (N2%cont = 0) Entonces
            suma = suma + cont
        Fin Si
        cont = cont + 1
    Fin Mientras
    Si (suma = N1) Entonces
        Escribir N1, "es amigo de ", N2
    Sino
        Escribir N1, "no es amigo de ", N2
    Fin Si
Si no
    Escribir N1, "no es amigo de ", N2
Fin Si
FIN ALGORITMO
```

---

```
/*
Determina si dos numeros son amiguitos
Relacion 2. Ej- 13
*/
#include <stdio.h>
main ()
{
    int N1, N2, cont=1, suma=0;
    system ("cls");
    printf ("Escriba numero A: ");
    scanf ("%d", &N1);
    printf ("Escriba numero B: ");
    scanf ("%d", &N2);
    while (cont < N1)
    {
        if (N1%cont == 0)
        {
            suma = suma + cont;
        }
        cont = cont + 1;
    }
    if (suma == N2)
    {
        suma = 0;
        cont = 1;
        while (cont < N2)
        {
            if (N2%cont == 0)
```

```
        {
            suma = suma + cont;
        }
        cont = cont + 1;
    }
    if (suma == N1)
    {
        printf ("%d es primito de %d\n", N1, N2);
    }
    else
    {
        printf ("%d no es primito de %d\n", N1, N2);
    }
}
else
{
    printf ("%d no es primito de %d\n", N1, N2);
}
}
```

**14.- Algoritmo que lee un número X y otro número entero positivo N y calcula la N\_ésima potencia de X.**

PROGRAMA:

poten.psc

ENTORNO:

Base, Expon, Resul -> Numeros enteros

ALGORITMO:

Leer Base

Leer Expon

Mientras (Expon < 0) Entonces

    Escribir "El exponente es erroneo, escriba otro: "

    Leer Expon

Fin Mientras

Resul = Base

Si (Expon = 0) Entonces

    Resul = 1

Sino

    Mientras (Expon > 1) Hacer

        Resul = Resul \* Base

        Expon = Expon - 1

    Fin Mientras

Fin Si

Escribir "La potencia pedida es => ", Resul

FIN ALGORITMO

---

/\*

Programa q calcula la potencia

Relacion 2. Ej- 14

```
*/
#include <stdio.h>
main ()
{
    int Base, Expon, Resul;
    printf ("Introduzca la base: ");
    scanf ("%d", &Base);
    printf ("Introduzca el exponente: ");
    scanf ("%d", &Expon);
    while (Expon < 0)
    {
        printf ("El Exponente es erroneo, escriba otro\n");
        printf ("que sea positivo: ");
        scanf ("%d", &Expon);
    }
    Resul = Base;
    if (Expon == 0)
    {
        Resul = 1;
    }
    else
    {
        while (Expon > 1)
        {
            Resul = Resul * Base;
            Expon = Expon - 1;
        }
    }
    printf ("La potencia pedida es => %d", Resul);
}
```

**15.- Algoritmo que lee 100 datos, compuesto cada uno de ellos por un nombre de persona y su sueldo neto mensual e imprime los datos del que mas cobra.**

PROGRAMA:

mascobra.psc

ENTORNO:

LNG -> Constante = 50

nombre[LNG], nMayor[LNG] -> Tablas de caracteres (cadena de texto)

sueldo, sMayor -> Numeros reales (coma flotante)

cont -> Numero entero

ALGORITMO:

sMayor = 0

cont = 0

Mientras (cont < 100) Hacer

Leer nombre

Leer sueldo

Si (sueldo > sMayor) Entonces

```
nMayor = nombre
sMayor = sueldo
Fin Si
cont = cont + 1
Fin Mientras
Escribir nMayor, "es el que mas cobra, con un sueldo de ", sMayor
FIN ALGORITMO
```

---

```
/* Relación 2. Ej - 15 */
#include <string.h>
#include <stdio.h>
#define LNG 50 /* LNG es la longitud de la cadena de texto */
main ()
{
    char nombre[LNG], nMayor[LNG];
    float sueldo, sMayor=0;
    int cont=0;
    while (cont < 100)
    {
        system ("cls");
        printf ("\n\n");
        printf ("Debe usted introducir 100 nombres con sus sueldos\n");
        printf ("Lleva %d introducidos\n", cont);
        printf ("NOMBRE => ");
        scanf ("%s", nombre);
        printf ("SUELDO => ");
        scanf ("%f", &sueldo);
        if (sueldo > sMayor)
        {
            strcpy(nMayor, nombre);
            sMayor = sueldo;
        }
        cont++;
    }
    printf ("%s es el que mas cobra, con un sueldo de %.0f ptas\n", nMayor, sMayor);
}
```

**16.- Algoritmo que recibe como dato de entrada dos números enteros positivos (N y M) y calcula los múltiplos de N menores que M.**

PROGRAMA:

pot2.psc

ENTORNO:

NuMayor, NuMenor, resul, cont -> Numeros enteros

ALGORITMO:

Leer NuMayor

Leer NuMenor

Si (NuMayor < NuMenor) Entonces

resul = NuMenor

```
NuMenor = NuMayor
NuMayor = resul
resul = 0
Fin Si
cont = 1
Mientras (resul < NuMayor) Hacer
    Escribir resul
    resul = cont * NuMenor
    cont = cont + 1
Fin Mientras
FIN ALGORITMO
```

---

```
/*
Calcula todos los multiplos menores
Relacion 2. Ej - 16.
*/
#include <stdio.h>
main ()
{
    int NuMayor, NuMenor, resul, cont;
    system ("cls");
    printf ("Escriba el numero mayor: ");
    scanf ("%d", &NuMayor);
    printf ("Escriba el numero menor: ");
    scanf ("%d", &NuMenor);
    if (NuMayor < NuMenor)
    {
        resul = NuMenor;
        NuMenor = NuMayor;
        NuMayor = resul;
        resul = 0;
    }
    cont = 1;
    while (resul < NuMayor)
    {
        printf ("%d\n",resul);
        resul= cont * NuMenor;
        cont = cont + 1;
    }
}
```

**17.- Algoritmo que recibe como dato de entada dos números enteros positivos (N y M) y calcula las potencias de N menores que M.**

PROGRAMA:

pot3.psc

ENTORNO:

NuMayor, NuMenor, resul, cont -> Numeros enteros

ALGORITMO

```
Leer NuMayor
Leer NuMenor
Si (NuMayor < NuMenor) Entonces
    resul = NuMenor
    NuMenor = NuMayor
    NuMayor = resul
Fin Si
resul = 1
Mientras (resul < NuMayor) Hacer
    Escribir resul
    resul = resul * NuMenor
Fin Mientras
FIN ALGORITMO
```

---

```
/*
Calcula todas las potencias menores
Relacion 2. Ej - 17.
*/
#include <stdio.h>
main ()
{
    int NuMayor, NuMenor, resul, cont;
    system ("cls");
    printf ("Escriba el numero mayor: ");
    scanf ("%d", &NuMayor);
    printf ("Escriba el numero menor: ");
    scanf ("%d", &NuMenor);
    if (NuMayor < NuMenor)
    {
        resul = NuMenor;
        NuMenor = NuMayor;
        NuMayor = resul;
    }
    resul = 1;
    while (resul < NuMayor)
    {
        printf ("%d\n",resul);
        resul= resul * NuMenor;
    }
}
```

**18.- Algoritmo que lee 50 nombres y escribe el número de veces que se repite el 1º de ellos.**

PROGRAMA:

repinom.psc

ENTORNO:

nombre, prime -> Cadenas de texto.

Nveces, cuenta -> Numeros enteros.



ALGORITMO:

Nveces = 1

cuenta = 1

Leer nombre

prime = nombre

Mientras (cuenta < 50) Hacer

    Leer nombre

    cuenta = cuenta + 1

    Si (nombre = prime) Entonces

        Nveces = Nveces + 1

    Fin Si

Fin Mientras

Escribir "El nombre", prime, "se ha repetido", Nveces, "veces."

FIN ALGORITMO

---

/\*

Relacion 2. Ej - 18

\*/

#include <stdio.h>

#include <string.h>

#define LNG 30

#define lng 29

main ()

{

    char nombre[LNG], prime[LNG];

    int Nveces=1, cuenta=1;

    int Si, N, exit, cont;

    system ("cls");

    printf ("Escriba un nombre terminado siempre en \_ => ");

    scanf ("%s", nombre);

    strcpy (prime, nombre);

    while (cuenta < 50)

    {

        system ("cls");

        printf ("Lleva usted %d nombres introducidos\n", cuenta);

        printf ("Introduzca otro, recuerde que debe terminar en\n");

        printf ("el caracter \_ => ");

        scanf ("%s", nombre);

        cuenta++;

        Si = 0; N=0; exit=0; cont=0;

        while (nombre[cont] != '\_' && prime[cont] != '\_')

        {

            cont++;

        }

        printf ("analizamos %d caracteres\n", cont);

        while (exit == 0)

        {

            if (N < cont)

```
    {
        if (nombre[N] == prime[N])
        {
            N++;
            Si = 1;
        }
        else
        {
            exit = 1;
            Si = 0;
        }
    }
    else
    {
        exit = 1;
    }
}
printf ("Hemos comprobado si las cadenas son iguales\n");
if (Si == 1)
{
    Nveces++;
}
}
printf ("El nombre %s se ha repetido %d veces\n", prime, Nveces);
}
```

**19.- Algoritmo que lee una secuencia de calificaciones numéricas (entre 0 y 100), que termina con el valor -1 y calcula e imprime la media aritmética, el número y porcentaje de aprobados y el número y porcentaje de suspensos.**

PROGRAMA:

estadis.psc

ENTORNO:

calif, contAp, contSu, contIn, contTo -> Numeros enteros

mediaAp, mediaSu, mediaAr -> Numeros reales (coma flotante)

ALGORITMO:

Leer calif

contAp = 0

contSu = 0

contIn = 0

contTo = calif + 1

Mientras (calif != -1) Hacer

    Si (calif > 49) Entonces

        contAp = contAp + 1

    Sino

        contSu = contSu + 1

    Fin Si

    Leer nota

    contIn = contIn + 1

```
    contTo = contTo + calif
Fin Mientras
Escribir "Ha introducido ", contIn, "notas."
mediaAr = contTo / contIn
Escribir "La media aritmética es: ", mediaAr
mediaAp = (contAp * 100) / contIn
Escribir "Han aprobado", contAp, "alumnos, siendo un", mediaAp, "% del total."
mediaSu = (contSu * 100) / contIn
Escribir "Han suspendido", contSu, "alumnos, siendo un", mediaSu, "% del total."
FIN ALGORITMO
```

---

```
/*
Estadística de calificaciones
Relación 2. Ej - 19.
*/
#include <stdio.h>
main ()
{
    int calif, ContAp, ContSu, ContIn, ContTo;
    float mediaAp, mediaSu, mediaAr;
    printf ("Escriba la nota. (Reuerde que para salir debe introducir -1)\n");
    printf ("Por favor, se ruega que las notas introducidas esten entre 0 y 100 => ");
    scanf ("%d", &calif);
    ContAp = 0;
    ContSu = 0;
    ContIn = 0;
    ContTo = calif+1;
    while (calif != -1)
    {
        if (calif > 49)
        {
            ContAp = ContAp + 1;
        }
        else
        {
            ContSu = ContSu + 1;
        }
        printf ("Esciba la nota => ");
        scanf ("%d", &calif);
        ContIn = ContIn + 1;
        ContTo = ContTo + calif;
    }
    printf ("Ha introducido %d Notas\n", ContIn);
    mediaAr = (float) ContTo / ContIn;
    printf ("La media aritmetica es => %.1f\n", mediaAr);
    mediaAp =(float) (ContAp * 100) / ContIn;
    printf ("Han aprobado %d alumnos, representando un %.3f %% del total.\n",
ContAp, mediaAp);
    mediaSu =(float) (ContSu * 100) / ContIn;
```

```
printf ("Han suspendido %d alumnos, representando un %.3f %% del total.\n",
ContSu, mediaSu);
}
```

**20.- Algoritmo que imprime la lista de los N primeros terminos de la sucesión de Fibonacci, siendo N el dato de entrada.**

PROGRAMA:

fibon.psc

ENTORNO:

Cantidad, N\_1, N\_2, N -> Numeros enteros

ALGORITMO:

N\_2 = 0

N\_1 = 1

Leer Cantidad

N = 1

Si (Cantidad = 1) Entonces

Escribir N\_2

Sino

Si (Cantidad = 2) Entonces

Escribir N\_2, N\_1

Sino

Escribir N\_2, N\_1, N

Cantidad = Cantidad - 3

Mientras (Cantidad > 0) Hacer

N\_2 = N\_1

N\_1 = N

N = N\_2 + N\_1

Escribir N

Cantidad = Cantidad - 1

Fin Mientras

Fin Si

Fin Si

FIN ALGORITMO

---

/\* relacion 2. Ej - 20. Sucesion de Fibonacci \*/

#include <stdio.h>

main()

{

int Cantidad, N\_1, N\_2, N;

system ("cls");

N\_2 = 0;

N\_1 = 1;

printf ("¿Cuántos terminos desea incluir en la sucesión => ");

scanf ("%d", &Cantidad);

N=1;

if (Cantidad == 1)

{

printf ("%d\n", N\_2);

```
    }
else
{
    if (Cantidad == 2)
    {
        printf ("%d\n%d\n", N_2, N_1);
    }
    else
    {
        printf ("%d\n%d\n%d\n", N_2, N_1, N_1);
        Cantidad = Cantidad - 3;
        while (Cantidad > 0)
        {
            N_2 = N_1;
            N_1 = N;
            N = N_1 + N_2;
            printf ("%d\n", N);
            Cantidad = Cantidad - 1;
        }
    }
}
}
```

```
printf("Numero 1 <=> Numero2 <=> Numero3 <=> Numero4\n");
```

```
printf (" %.0f      <=> %.0f      <=> %.0f      <=> %.0f \n",
        tabla[0][0], tabla[0][1], tabla[0][2], tabla[0][3]);
printf (" %.3f <=> %.3f <=> %.3f <=> %.3f \n",
        tabla[1][0], tabla[1][1], tabla[1][2], tabla[1][3]);
printf (" %.3f <=> %.3f <=> %.3f <=> %.3f \n",
        tabla[2][0], tabla[2][1], tabla[2][2], tabla[2][3]);
printf (" %.3f <=> %.3f <=> %.3f <=> %.3f \n",
        tabla[3][0], tabla[3][1], tabla[3][2], tabla[3][3]);
printf (" %.3f <=> %.3f <=> %.3f <=> %.3f \n",
        tabla[4][0], tabla[4][1], tabla[4][2], tabla[4][3]);
}
```

**22.- En la declaración sobre el impuesto de la renta, una persona tiene una deducción de 25.000 ptas por hijo y una devolución extra de 35.000 ptas si es mayor de 65 años. Dado el número de hijos y la edad del padre, calcular la deducción.**

PROGRAMA:

renta.psc

ENTORNO:

deduccion -> Numero en coma flotante

Nhijos, Edad -> Numeros enteros

ALGORITMO:

Leer Nhijos, Edad

Si (Edad < 65) Entonces

deduccion = Nhijos \* 25000

Sino

Deduccion = 35000 + (Nhijos \* 25000)

Fin Si

Escribir deduccion

FIN ALGORITMO

---

/\*

Calcula la deduccion de la persona.

Ej - 22. Relacion 2B.

Por Victor Campuzano y Laura Gomez

\*/

#include <stdio.h>

main ()

{

float Deduccion=0;

int Nhijos, Edad;

system ("cls");

printf ("\n\n");

printf ("Deme usted su edad => ");

scanf ("%d", &Edad);

printf ("\nDeme usted su numero de hijos => ");

scanf ("%d", &Nhijos);

if (Edad < 65)

```
{
    Deduccion = Nhijos * 25;
}
else
{
    Deduccion = 35 + (Nhijos * 25);
}
printf ("\nLa deduccion a percibir es de => %.0f000", Deduccion);
}
```

**23.- Un agente de ventas puede tener las siguientes comisiones:**

- a) Si el importe de la venta es mayor de 50.000 ptas, la comisión es 5.000 ptas mas el 8% de lo que excede el importe de 50.000 ptas.**
- b) Si  $5.000 \leq \text{importe} < 50.000$ , la comisión es del 10%.**
- c) Si el importe es menor que 5.000 ptas, no hay comisión.**

**Escribir un programa que lea el importe total y calcule e imprima la comisión.**

PROGRAMA:

comision.psc

ENTORNO:

Importe, Comision -> Numeros enteros (long int)

ALGORITMO:

Leer Importe

Si (Importe < 5000) Entonces

comision = 0

Si no

Si (Importe < 50000) Entonces

Comision = Importe \* 0.1

Sino

Comision = 5000 + ((Importe - 50000) \* 0.08)

Fin Si

Fin Si

Escribir Comision

FIN ALGORITMO

---

/\*

Calcula la comision del agente de Ventas.

Ej - 23. Relacion 2B.

Por Victor Campuzano y Laura Gomez

\*/

#include <stdio.h>

main ()

{

long int Importe, Comision;

system ("cls");

printf ("\n\n");

printf ("Deme usted el importe => ");

scanf ("%ld", &Importe);

if (Importe < 5000)



```
{
    Comision = 0;
}
else
{
    if (Importe < 50000)
    {
        Comision = Importe * 0.1;
    }
    else
    {
        Comision = 5000 + ((Importe - 50000) * 0.08);
    }
}
printf ("\n\n");
printf ("La comision a cobrar es: %ld\n", Comision);
}
```

**24.- Escribe un programa que calcule la suma  $1/(2n-1)$  desde 1 hasta un valor de entrada.**

PROGRAMA:

ej24.psc

ENTORNO:

cont, N -> Numeros enteros

suma, resul -> Numeros en coma flotante

ALGORITMO:

suma =0

Leer N

Para (cont=1 hasta cont=N con incremento de 1) Hacer

    resul =  $1/((2*cont)-1)$

    suma = suma + resul

Fin Para

Escribir suma

FIN ALGORITMO

---

/\*

Calcula el sumatorio  $1/(2n-1)$ .

Ej - 24. Relacion 2B

Por Victor Campuzano y Laura Gomez

\*/

#include <stdio.h>

main()

{

    int cont, N;

    float suma=0, resul;

    system ("cls");

    printf ("\n\nEscriba un numero => ");

    scanf ("%d", &N);

```
for (cont=1;cont<=N;cont++)
{
    resul =(float) 1/((2*cont)-1);
    suma = suma + resul;
}
printf ("\nLa suma es => %.3f\n", suma);
}
```

**25.- Hacer un programa de conversión para los ingleses. Introducida una longitud en metros, se convierte en Yardas, Pies y Pulgadas.**

PROGRAMA:

longitudes.psc

ENTRNO:

yardas, pies, pulgadas -> Numeros enteros

metros -> Numero en coma flotante

longi -> Numero entero

ALGORITMO:

yardas =0

pies =0

pulgadas =0

Leer longi

metros = longi

Mientras (metros >= 0.9144) Hacer

    metros = metros - 0.9144

    yardas = yardas + 1

Fin Mientras

Mientras (metros >= 0.3048) Hacer

    metros = metros - 0.3048

    pies = pies + 1

Fin Mientras

Mientras (metros > 0) Hacer

    metros = metros - 0.0254

    pulgadas = pulgadas + 1

Fin Mientras

Escribir yardas, "+" pies, "+" pulgadas

FIN ALGORITMO

---

/\* Conversion de longitud para los ingleses

Ej - 25. Relacion 2B

Por Victor Campuzano Gallego y Laura Gomez Carrillo\*/

#include <stdio.h>

main()

{

    int yardas=0, pies=0, pulgadas=0;

    float metros;

    int Longi;

    system ("cls");

    printf ("\n\nDeme usted la longitud en metros => ");

```
scanf ("%d", &Longi);
metros = Longi;
while (metros>=0.9144)
{
    metros = metros - 0.9144;
    yardas = yardas + 1;
}
while (metros>=0.3048)
{
    metros = metros - 0.3048;
    pies = pies + 1;
}
while (metros>0)
{
    metros = metros - 0.0254;
    pulgadas = pulgadas + 1;
}
printf ("\n%d metros es: %d Yardas + %d Pies + %d Pulgadas\n",
        Longi, yardas, pies, pulgadas);
printf ("Sobran %.4f metros\n", metros);
}
```

**26.- Un comerciante dispone de tres tipos de cables. Dado el tipo de cable y la longitud, calcular e imprimir el importe. (Tipo 1: 24 ptas/m; Tipo 2: 37 ptas/m; Tipo 3: 52 ptas/m).**

PROGRAMA:

cables.psc

ENTORNO:

tipo, longitud, precio -> Numeros enteros  
importe -> Numero entero (long int)

ALGORITMO:

Leer tipo

Leer longitud

Si (tipo = 1) Entonces

    precio = 24

Sino

    Si (tipo = 2) Entonces

        precio = 37

    Sino

        precio = 52

    Fin Si

Fin Si

importe = longitud \* precio

Escribir importe

FIN ALGORITMO

---

/\* Calcula el precio del cable.

Ej - 26. Relacion 2B

Por Victor Campuzano Gallego y Laura Gomez Carrillo\*/

```
#include <stdio.h>
main ()
{
    int tipo, longitud, precio;
    long int importe;
    system ("cls");
    printf ("\n\nDeme usted el tipo de cable: ");
    scanf ("%d", &tipo);
    while (tipo<1 || tipo>3)
    {
        system ("cls");
        printf ("\n\nEl tipo es erroneo, intentelo de nuevo: ");
        scanf ("%d", &tipo);
    }
    printf ("\n\nDeme usted el N de metros: ");
    scanf ("%d", &longitud);
    if (tipo == 1)
    {
        precio = 24;
    }
    else
    {
        if (tipo == 2)
        {
            precio = 37;
        }
        else
        {
            precio = 52;
        }
    }
    importe = longitud * precio;
    printf ("\n\n%d metros de tipo %d cuestan %ld ptas\n", longitud, tipo, importe);
}
```

**27.- Algoritmo que permita calcular el número máximo de términos de la sucesión de Fibonacci cuya suma es menor de 250.**

PROGRAMA:

fibon2.psc

ENTORNO:

Cantidad, N\_1, N\_2, N -> Numeros enteros

ALGORITMO:

N\_2 = 0

N\_1 = 1

N = 1

Escribir N\_2, N\_1, N

Cantidad = 2

Mientras (Cantidad < 200) Hacer

    N\_2 = N\_1

    N\_1 = N

    N = N\_2 + N\_1

    Cantidad = Cantidad + N

    Escribir N

Fin Mientras

FIN ALGORITMO

---

/\* relacion 2B. Ej - 27. Sucesion de Fibonacci. Parte II \*/

#include <stdio.h>

main()

{

    int Cantidad, N\_1, N\_2, N;

    system ("cls");

    N\_2 = 0;

    N\_1 = 1;

    N=1;

    printf ("%d\n%d\n%d\n", N\_2, N\_1, N\_1);

    Cantidad = 2;

    while (Cantidad < 200)

    {

        N\_2 = N\_1;

        N\_1 = N;

        N = N\_1 + N\_2;

        Cantidad = Cantidad + N;

        printf ("%d === %d\n", N, Cantidad);

    }

}

**28.- Escribir un programa para simplificar la fracción M/N, siendo éstos números enteros.**

PROGRAMA:

    simplif.psc

ENTORNO:

    numerador, denominador, cont -> Numeros enteros

ALGORITMO:

cont = 2

Leer numerador

Leer denominador

Mientras (cont<=numerador Y cont<= denominador) Hacer

    Si (numerador%cont=0 Y denominador%cont=0) Entonces

        numerador = numerador / cont

        denominadro = denominador / cont

    Fin Si

    cont = cont + 1

Fin Mientras  
Escribir numerador, denominador  
FIN ALGORITMO

---

```
/* Simplifica dos fracciones.  
Ej - 28. Relacion 2B.  
Por Victor Campuzano Gallego y Laura Gomez Carrillo*/  
#include <stdio.h>  
main ()  
{  
    int numerador, denominador, cont=2;  
    system ("cls");  
    printf ("\n\nEscriba el Numerador: ");  
    scanf ("%d", &numerador);  
    printf ("\nEscriba el Denominador: ");  
    scanf ("%d", &denominador);  
    while (cont<=numerador && cont<=denominador)  
    {  
        if (numerador%cont==0 && denominador%cont==0)  
        {  
            numerador = numerador / cont;  
            denominador = denominador / cont;  
        }  
        cont++;  
    }  
    printf ("\nLa fraccion simplificada es %d / %d\n", numerador, denominador);  
}
```

**29.- Escribir un programa que calcule y escriba las parejas de números primos gemelos inferiores a un número N de entrada.**

PROGRAMA:

gemelos.psc

ENTORNO:

auxiliar, auxiliar2, numero -> Numeros enteros

pareja, primo, cont -> Numeros enteros

ALGORITMO:

Leer numero

auxiliar = 1

auxiliar2 = 3

Mientras (numero>auxiliar+2) Hacer

    primo = 1

    cont = 2

    Mientras (cont<auxiliar) Hacer

        Si (auxiliar&cont=0) Entonces

            primo = 0

        Fin Si

        cont = cont + 1

    Fin Mientras

```
Si (primo = 1)
    pareja = auxiliar + 2
    cont = 2
    Mientras (cont<pareja) Hacer
        Si (pareja&contador=0) Entonces
            primo = 0
        Fin Si
        cont = cont + 1
    Fin Mientras
    Si (primo = 1) Entonces
        Escribir auxiliar, "y", pareja
    Fin Si
    pareja = pareja + 1
Fin Si
auxiliar = auxiliar + 1
Fin Mientras
Si (numero<=auxiliar) Entonces
    Escribir ("No hay parejas")
Fin Si
FIN ALGORITMO
```

---

```
/*Numeros Gemelos
Relacion 2B. Ej - 29
Por Victor Campuzano Gallego y Laura Gomez Carrillo*/
#include<stdio.h>
main()
{
    int auxiliar,auxiliar2,numero,pareja,primo,contador;
    printf("\nDame un número: ");
    scanf("%d",&numero);
    printf("\nLas parejas de n$primos gemelos menores a %d son:",numero);
    auxiliar=1;
    auxiliar2=3;
    while(numero>auxiliar+2)
    {
        primo=1;
        contador=2;
        while(contador<auxiliar)
        {
            if(auxiliar%contador==0)
            {
                primo=0;
            }
            contador=contador+1;
        }
        if(primo==1)
        {
            pareja=auxiliar+2;
            contador=2;
        }
    }
}
```

```
while(contador<pareja)
{
    if(pareja%contador==0)
    {
        primo=0;
    }
    contador=contador+1;
}
if(primo==1)
{
    printf("\nEl %d y %d",auxiliar,pareja);
}
pareja=pareja+1;
}
auxiliar=auxiliar+1;
}
if(numero<=auxiliar2)
{
    printf("\nNo hay parejas\n");
}
}
```

**30.- Escribe un programa que lea números enteros y positivos y que calcule cuantos son pares, los múltiplos de 3, de 5, y de 7 que existan en los datos de entrada.**

PROGRAMA:

ej30.psc

ENTORNO:

numero -> Numero entero

ALGORITMO:

Leer numero

Si (numero%2=0) Entonces

Escribir "Es par"

Fin Si

Si (numero%3=0) Entonces

Escribir "Es multiplo de 3"

else

Si (numero%5 = 0) Entonces

Escribir "Es multiplo de 5"

Si no

Si (numero%7 =0) Entonces

Escribir "Es multiplo de 7"

Fin Si

Fin Si

Fin Si

FIN ALGORITMO

---

/\*Calcula los pares y los multiplos



Relacion 2B. Ej - 30

```
*/  
#include<stdio.h>  
main()  
{  
    int numero;  
    printf("\nDame un número: ");  
    scanf("%d",&numero);  
    if(numero%2==0)  
    {  
        printf("\nHas escrito un numero par");  
    }  
    if(numero%3==0)  
    {  
        printf("\nEl %d es multiplo de 3.",numero);  
    }  
    else  
    {  
        if(numero%5==0)  
        {  
            printf("\nEl %d es multiplo de 5.",numero);  
        }  
        else  
        {  
            if(numero%7==0)  
            {  
                printf("\nEl %d es multiplo de 7.",numero);  
            }  
        }  
    }  
}
```

**31.- Diseñar un algoritmo correspondiente a un programa que calcule la suma de los 20 primeros números pares y el producto de los 20 primeros números impares, simultáneamente.**

PROGRAMA:

simult.psc

ENTORNO:

cont, num -> Numeros enteros

pares, impares -> Numeros enteros (long int)

ALGORITMO:

cont =1

num=1

pares=0

impares=1

Mientras (cont <=20) Hacer

    impares = impares \* num

    num = num + 1

```
pares = pares + num
num = num + 1
Fin Mientras
Escribir pares
Escribir impares
FIN ALGORITMO
```

---

```
/* Calculos simultaneos.
Relacion 2B. Ej - 31.
Por Victor Campuzano Gallego y Laura Gomez Carrillo */
#include <stdio.h>
main()
{
    int cont=1, num=1;
    long int pares=0, impares=1;
    while (cont <= 20)
    {
        impares = impares * num;
        num++;
        pares = pares + num;
        num++;
        cont++;
    }
    printf("\nLa suma de los 20 primeros pares es: %ld\n", pares);
    printf("\nEl producto de los 20 primeros impares es: %ld\n", impares);
}
```

**32.- Diseñar el algoritmo correspondiente a un programa que lee dos números y presenta opciones correspondientes a distintas operaciones que podemos realizar con ellos (suma, resta, multiplicación, división) de manera que en función de la opción elegida, muestra el resultado de la operación realizada. En aquellos casos en los que se desee seguir operando con los mismos números, se debe contestar ‘S’ al siguiente mensaje: “¿Otra operación con los mismos números S/N?”. Finalmente, el programa deberá concluir tras responder ‘S’ a la pregunta de salida.**

SUBPROGRAMA:

oper

RECIBE:

a, b -> Numeros enteros

DEVUELVE:

Numero en coma flotante

ENTORNO:

opcion -> Numero entero

ALGORITMO:

Leer opcion

Si (opcion = 1) Entonces

devuelve a+b

Sino

```
Si (opcion = 2) Entonces
    devuelve a-b
Sino
    Si (opcion = 3) Entonces
        devuelve a*b
    Si no
        devuelve a/b
    Fin Si
Fin Si
Fin Si
FIN SUBPROGRAMA
PROGRAMA:
    dosnum.psc
ENTORNO:
    num_a, num_b -> Numeros enteros
    fin, elige -> Caracteres
    resultado -> Numero en coma flotante
ALGORITMO:
fin = 'N'
elige = 'S'
Mientras (fin = 'N') Hacer
    Leer num_a
    Leer num_b
    elige = 'S'
    Mientras (elige = 'S') Hacer
        resultado = oper(num_a, num_b)
        Escribir resultado
        Leer elige
    Fin Mientras
    Leer fin
Fin Mientras
FIN ALGORITMO
```

---

/\* Operaciones con dos numeros.

Relacion 2B. Ej - 32.

Por Victor Campuzano Gallego y Laura Gomez Carrillo \*/

#include <stdio.h>

```
float oper(int a, int b)
{
    int opcion;
    system ("cls");
    printf("\n\nQue operacion desea realizar con estos numeros?\n");
    printf("1.-Suma\n2.-Resta\n3.-Multiplicacion\n4.-Division\n");
    scanf ("%d", &opcion);
    if (opcion == 1)
    {
        return a+b;
    }
}
```

```
else
{
    if (opcion == 2)
    {
        return a-b;
    }
    else
    {
        if (opcion == 3)
        {
            return a*b;
        }
        else
        {
            return (float)a/b;
        }
    }
}
}
main ()
{
    int num_a, num_b;
    char fin='N', elige='S';
    float resultado;
    while (fin == 'N')
    {
        system ("cls");
        printf ("\n\nEscriba un numero: ");
        scanf ("%d", &num_a);
        printf ("\nEscriba otro numero: ");
        scanf ("%d", &num_b);
        elige = 'S';
        while (elige == 'S')
        {
            resultado = oper(num_a, num_b);
            printf ("\nEl resultado es: %.3f\n", resultado);
            printf ("\n¿Desea realizar otra operación con estos numeros: ");
            scanf ("%c", &elige);
        }
        printf ("\n¿Esta seguro de que desea salir?: ");
        scanf ("%c", &fin);
    }
}
```

**33.- Diseñar un programa que permita calcular las gratificaciones a pagar a los empleados de una empresa con motivo de celebrar sus 20 años de funcionamiento. Se deberá leer el nombre de la persona y su edad en la empresa. La entrada de datos finaliza al introducir un nombre igual a 'FIN'. Se desea**

realizar un listado con los nombres de los empleados y su gratificación correspondiente, teniendo en cuenta:

- a) Mas de 3 y menos de 10 años → 50.000
- b) Entre 10 y 15 años → 75.000
- c) Mas de 15 años → 110.000

Al final, se desea escribir el total de las gratificaciones.

PROGRAMA:

gratifi.psc

ENTORNO:

CANT=5, LNG=30 -> Constantes

nombres[CANT][LNG] -> tabla de caracteres

datos[CANT][2] -> tabla de enteros (long int)

cont -> Numero entero

nada -> Caracter

suma -> Numero entero (long int)

ALGORITMO:

suma =0

Para (cont=0 hasta cont=CANT-1 con incremento de 1) Hacer

Leer nombres[cont]

Leer datos[cont][0]

Si (datos[cont][0]>3 Y datos[cont][0]<10) Entonces

datos[cont][1] = 50000

Sino

Si (datos[cont][0]<= 15 Y datos[cont][0]>=10) Entonces

datos[cont][1] = 75000

Si no

Si (datos[cont][0]>15) Entonces

datos[cont][1] = 110000

Sino

datos[cont][1] = 0

Fin Si

Fin Si

Fin Si

suma = suma + datos[cont][1]

Fin Para

Para (cont=0 hasta cont=CANT con incremento de 1) Hacer

Escribir "NOMBRE=>", nombres[cont]

Escribir "EDAD =>", datos[cont][0]

Escribir "GRATIFICACION =>", datos[cont][1]

Fin Para

Escribir suma

FIN ALGORITMO

/\*

Calcula las gratificaciones.

Ej - 33. Relacion 2B

Por Victor Campuzano Gallego y Laura Gomez Carrillo

\*/

```
#include <stdio.h>

#define CANT 5 /*Cantidad de nombres a registrar en la tabla*/
#define LNG 30 /*Longitud max de la cadena de texto */

main ()
{
    char nombres[CANT][LNG];
    long int datos[CANT][2], suma=0; /* Edad y Gratificacion */
    int cont;
    char nada;
    for (cont=0; cont<CANT; cont++)
    {
        system ("cls");
        printf ("\n\nEscriba el nombre => ");
        scanf ("%s", nombres[cont]);
        printf ("\nEscriba la edad => ");
        scanf ("%ld", &datos[cont][0]);
        if (datos[cont][0]>3 && datos[cont][0]<10)
        {
            datos[cont][1] = 50000;
        }
        else
        {
            if (datos[cont][0]<=15 && datos[cont][0]>=10)
            {
                datos[cont][1] = 75000;
            }
            else
            {
                if (datos[cont][0]>15)
                {
                    datos[cont][1] = 110000;
                }
                else
                {
                    datos[cont][1] = 0;
                }
            }
        }
        suma = suma + datos[cont][1];
    }
    system ("cls");
    for (cont=0; cont<CANT; cont++)
    {
        printf ("\nNOMBRE => %s", nombres[cont]);
        printf ("\nEDAD  => %ld", datos[cont][0]);
        printf ("\nGRATIFICACION => %ld", datos[cont][1]);
        printf ("\n*****");
    }
}
```

```
printf("\n¿Desea ver el siguiente? (s = SI) => ");
scanf ("%c", &nada);
if (nada == 's')
{
    system ("cls");
}
else
{
    cont = CANT;
}
}
printf ("\n*****\n");
printf ("\n*****\n");
printf ("TOTAL GRATIFICACIONES = %ld\n", suma);
}
```