

1. DATOS INFORMATIVOS

Carrera: Electronica y Automatización

Asignatura: Fundamentos de la Programación

Docente: RUIZ ROBALINO JENNY ALEXANDRA

Nombre: Ederson Gualoto

Ejercicios (Unidad 1)

Instrucciones Generales

1. Lea atentamente cada enunciado del ejercicio.
2. Complete las tablas indicadas con los valores solicitados.
3. Realice el análisis de entradas, procesos y salidas (E/P/S) según corresponda.
4. Todos los ejercicios deben resolverse en pseudocódigo o diagrama de flujo (según indique el docente).

Ejercicio 1 (Media de dos valores)

Desarrolle un programa que lea dos números reales desde teclado e imprima su media. los requisitos funcionales son: primero se debe leer un valor real y se guarda en una variable X; a continuación, se lee y almacena el Segundo valor en Y, Finalmente la variable RES recibe la media de ambos valores y los muestra por pantalla.

OBJETO	Nombre	Valor	Tipo
Valor	X	Variable	Real
Valor	Y	Variable	Real
Media valores	RES	Variable	Real
Dos	2	Constante	Entero

```
#include <stdio.h>

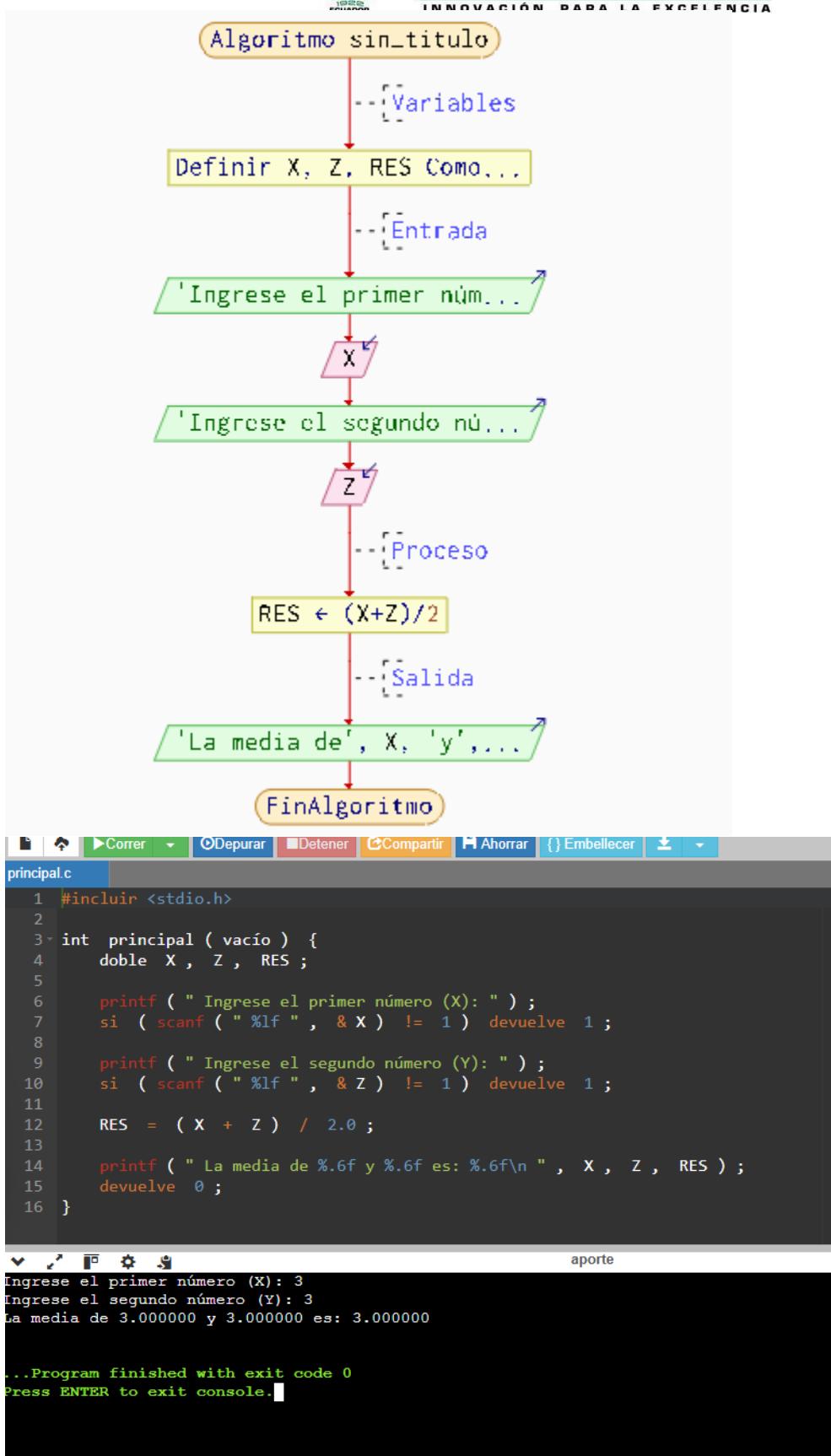
int main(void) {
    double X, Z, RES;

    printf("Ingrese el primer número (X): ");
    if (scanf("%lf", &X) != 1) return 1;

    printf("Ingrese el segundo número (Y): ");
    if (scanf("%lf", &Z) != 1) return 1;

    RES = (X + Z) / 2.0;

    printf("La media de %.6f y %.6f es: %.6f\n", X, Z, RES);
    return 0;
}
```



Ejercicio 2 (Valor absoluto de X al cubo)

Desarrolle un programa que lea un número real X y escriba por pantalla $|X|^3$

Variable	Nombre	Valor	Tipo
Numero	X	Variable	Real
Valor absoluto	XABS	Variable	Real
Resultado	Res	Variable	Real
Numero	3	Constante	Entero

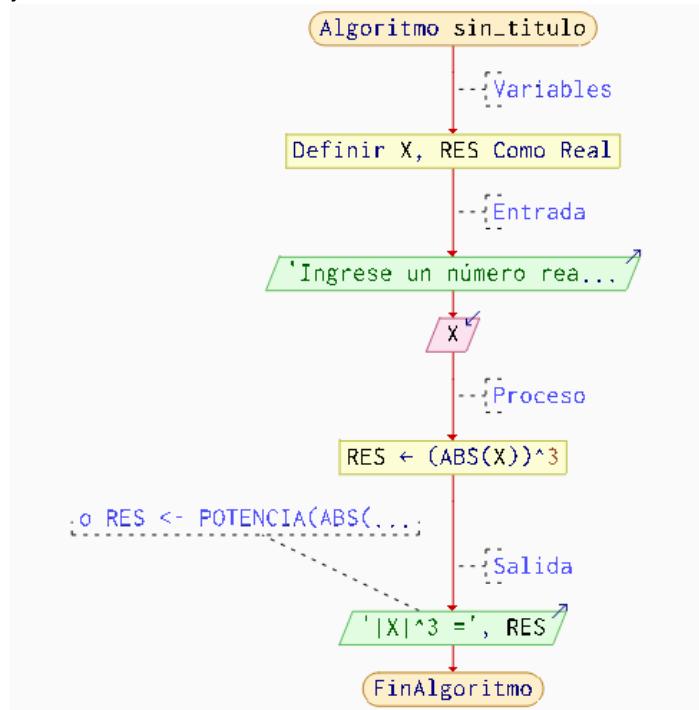
```
#include <stdio.h>
#include <math.h>

int main(void) {
    double X, A, RES;

    printf("Ingrese un número real X: ");
    if (scanf("%lf", &X) != 1) return 1;

    A = fabs(X);      // valor absoluto de X
    RES = A * A * A; // |X|^3

    printf("|\%.6f|^3 = %.6f\n", X, RES);
    return 0;
}
```



```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main(void) {
5     double X, A, RES;
6
7     printf("Ingrese un número real X: ");
8     if (scanf("%lf", &X) != 1) return 1;
9
10    A = fabs(X);           // valor absoluto de X
11    RES = A * A * A;      // |X|^3
12
13    printf("|%.6f|^3 = %.6f\n", X, RES);
14    return 0;
15 }
```

```
Ingrese un número real X: 4
|4.000000|^3 = 64.000000
```

```
...Program finished with exit code 0
Press ENTER to exit console.
```

Ejercicio 3 (Divisible)

Desarrolle un programa que lea dos números enteros por teclado y determine si el primero de ellos es divisible por el Segundo. Se mostrará en pantalla el resultado. Utilice el operador modulo % que devuelve el resto de la división.

Variable	Nombre	Valor	Tipo
Numérica	a	a>←0	Realse
Numérica	b	b>←0	Reales
Numérica	c	c>←0	Reales

```
#include <stdio.h>

int main(void) {
    int X, Y;

    printf("Ingrese el primer número entero (X): ");
    if (scanf("%d", &X) != 1) {
        fprintf(stderr, "Entrada inválida para X\n");
        return 1;
    }

    printf("Ingrese el segundo número entero (Y): ");
    if (scanf("%d", &Y) != 1) {
        fprintf(stderr, "Entrada inválida para Y\n");
        return 1;
    }
}
```

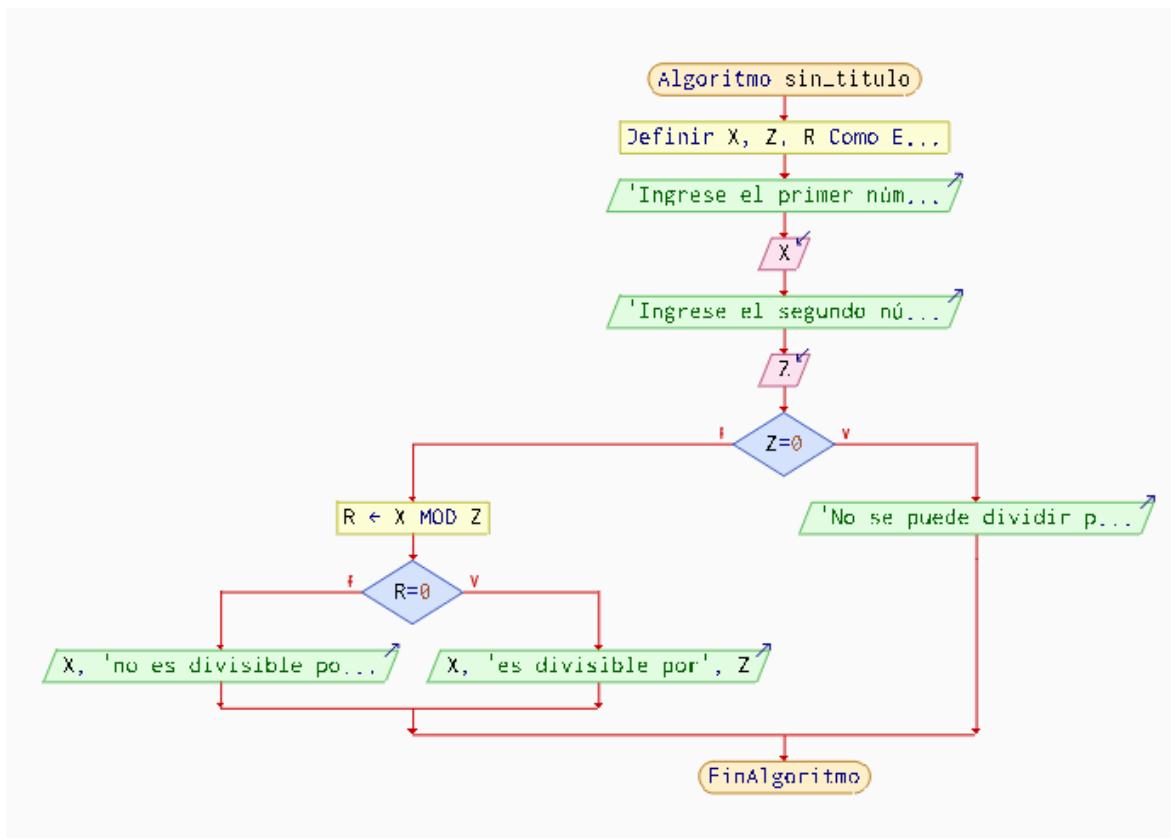
```

if (Y == 0) {
    printf("Error: división por cero no está definida.\n");
    return 0;
}

if (X % Y == 0)
    printf("%d es divisible por %d\n", X, Y);
else
    printf("%d no es divisible por %d (resto = %d)\n", X, Y, X % Y);

return 0;
}

```



principal.c

```

3 int principal () vacío ) {
4     entero X , Y ;
5
6     printf ( " Ingrese el primer número entero (X): " );
7     si ( scanf ( "%d " , & X ) != 1 ) {
8         fprintf ( stderr , " Entrada inválida para X \n " );
9         devuelve 1 ;
10
11    printf ( " Ingrese el segundo número entero (Y): " );
12    si ( scanf ( "%d " , & Y ) != 1 ) {
13        fprintf ( stderr , " Entrada inválida para Y \n " );
14        devuelve 1 ;

```

aporte

```

Ingrrese el primer número entero (X): 4
Ingrrese el segundo número entero (Y): 2
4 es divisible por 2

...Program finished with exit code 0
Press ENTER to exit console.

```

Ejercicio 4 (Intervalo)

Desarrolle un programa que lea un número real del teclado y determine si pertenece al Intervalo de (0 a 10), indicando por pantalla el resultado.

El número X introducido ha de cumplir $|X>0$ y $X \leq 10$ (ambas condiciones a la vez) para pertenecer al intervalo

Variable	Nombre	Valor	Tipo
Valor	X	Variable	Real
Límite 1	0	Constante	Entero
Límite 2	10	Constante	Entero
Mensaje	MSG	VARIABLE	Cadena

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

```

int main(void) {
    double X;

    printf("Ingrese un número real X: ");
    if (scanf("%lf", &X) != 1) {

```

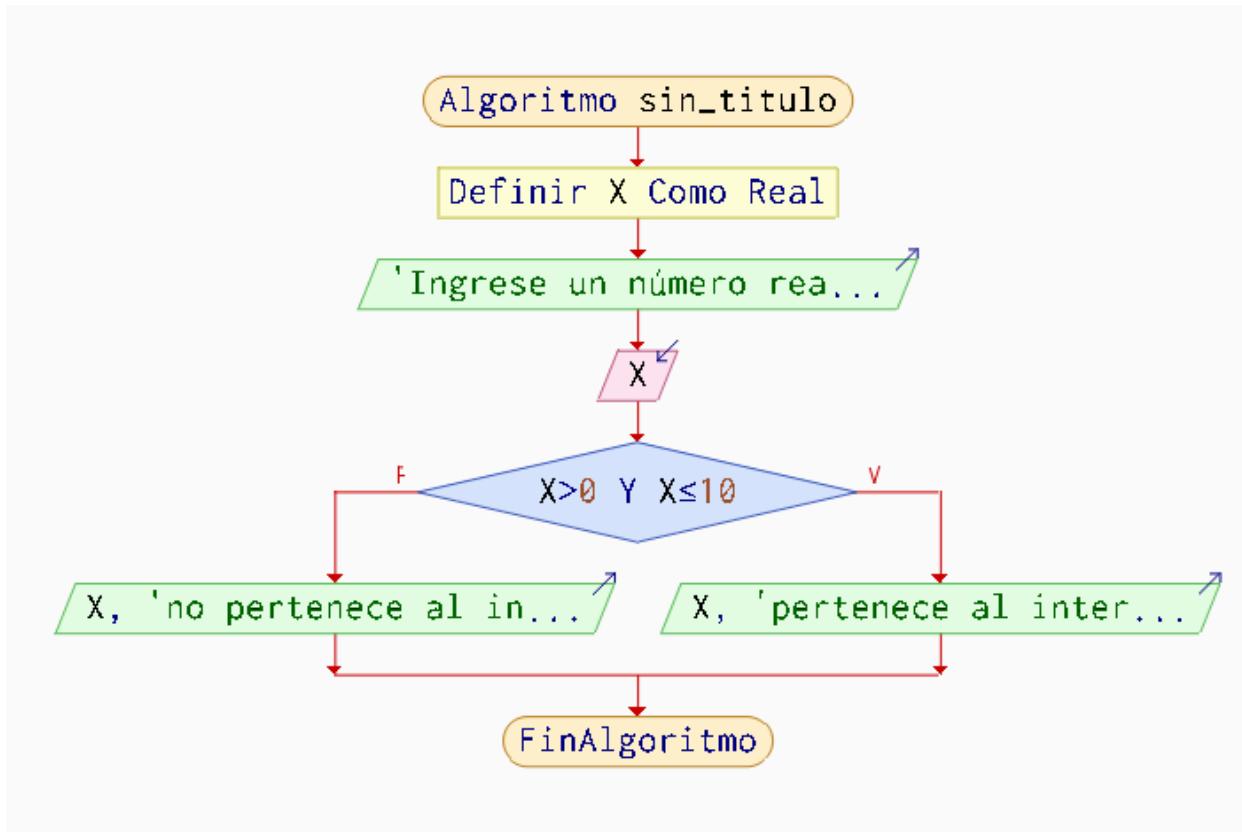
```

fprintf(stderr, "Entrada inválida\n");
return 1;
}

if (X > 0.0 && X <= 10.0)
    printf("%.6f pertenece al intervalo (0, 10]\n", X);
else
    printf("%.6f no pertenece al intervalo (0, 10]\n", X);

return 0;
}

```



principal.c

```

1 #incluir <stdio.h>
2
3 int principal () vacío {
4     doble X ;
5
6     printf ( " Ingrese un número real X: " ) ;
7     si ( scanf ( "%lf " , & X ) != 1 ) {
8         fprintf ( stderr , " Entrada inválida \n " ) ;
9         devuelve 1 ;
10    }
11
12    si ( X > 0.0 y X <= 10.0 )
13        printf ( "% .6f pertenece al intervalo (0, 10] \n " , X ) ;
14    demás
15        printf ( "% .6f no pertenece al intervalo (0, 10] \n " , X ) ;
16
17    devuelve 0 ;
18
19

```

Ingrese un número real X: 4
4.000000 pertenece al intervalo (0, 10]

...Program finished with exit code 0
Press ENTER to exit console.

Ejercicio 5(conversion de unidades de tiempo)

Desarrolle un programa que lea por teclado un valor entero X e, interpretando este valor como el número de segundos que dura un evento, calcule y muestre por pantalla cuántos días, horas, minutos y segundos representa X.

Variable	Nombre	Valor	Tipo
Dias	dias	variable	entero
Horas	horas	variable	entero
Minutos	minutos	variable	entero
Segundos	segundos	variable	entero
numero	60	variable	entero

```
#include <stdio.h>

int main(void) {
    long long X;

    printf("Ingrese el número entero de segundos: ");
    if (scanf("%lld", &X) != 1) {
        fprintf(stderr, "Entrada inválida\n");
        return 1;
    }

    if (X < 0) {
        printf("Entrada inválida: el número de segundos no puede ser negativo.\n");
    }
}
```

```

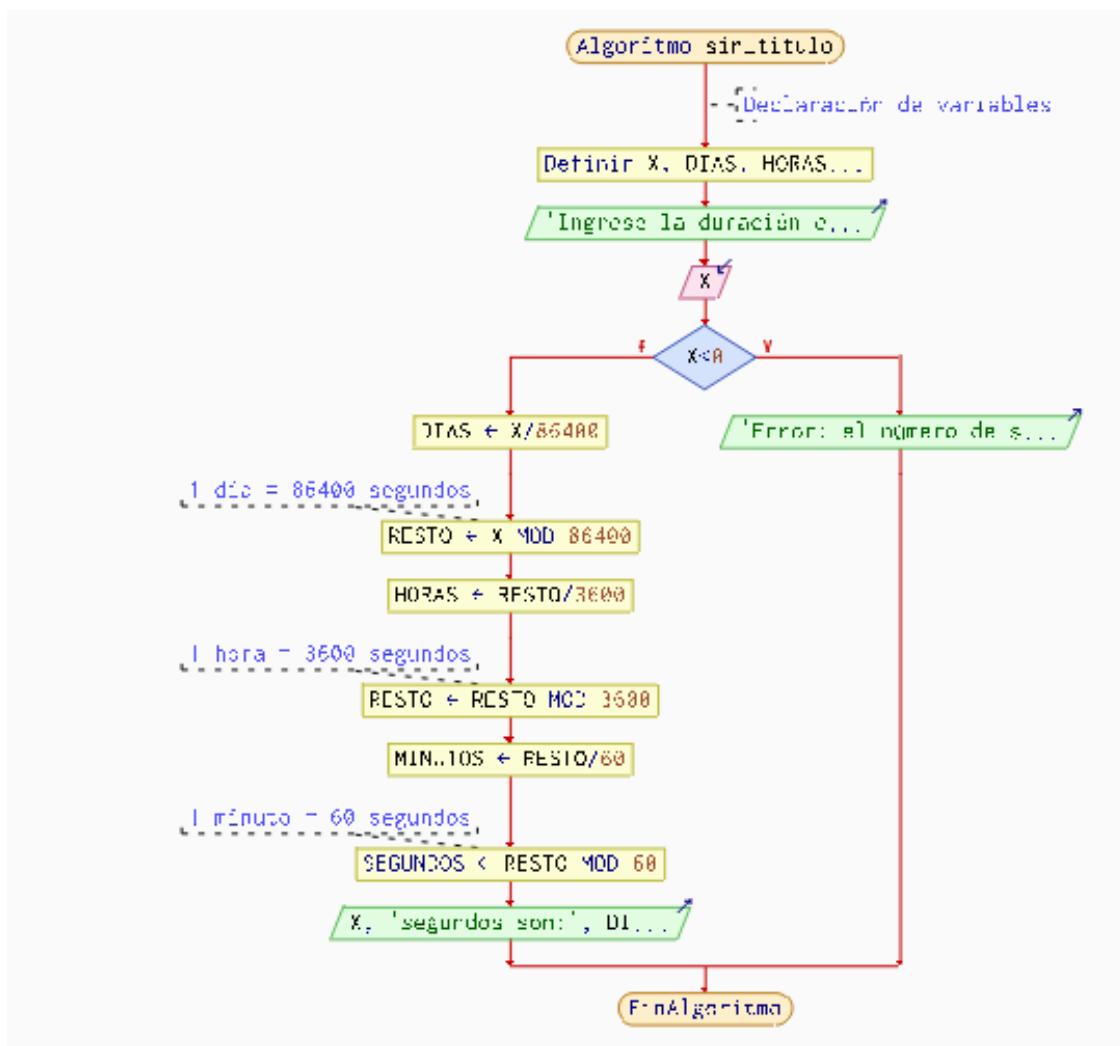
    return 1;
}

long long dias = X / 86400; // 1 día = 86400 s
long long resto = X % 86400;
long long horas = resto / 3600; // 1 hora = 3600 s
resto = resto % 3600;
long long minutos = resto / 60; // 1 minuto = 60 s
long long segundos= resto % 60;

printf("%lld segundos equivalen a: %lld días, %lld horas, %lld minutos y %lld segundos\n",
      X, dias, horas, minutos, segundos);

return 0;

```



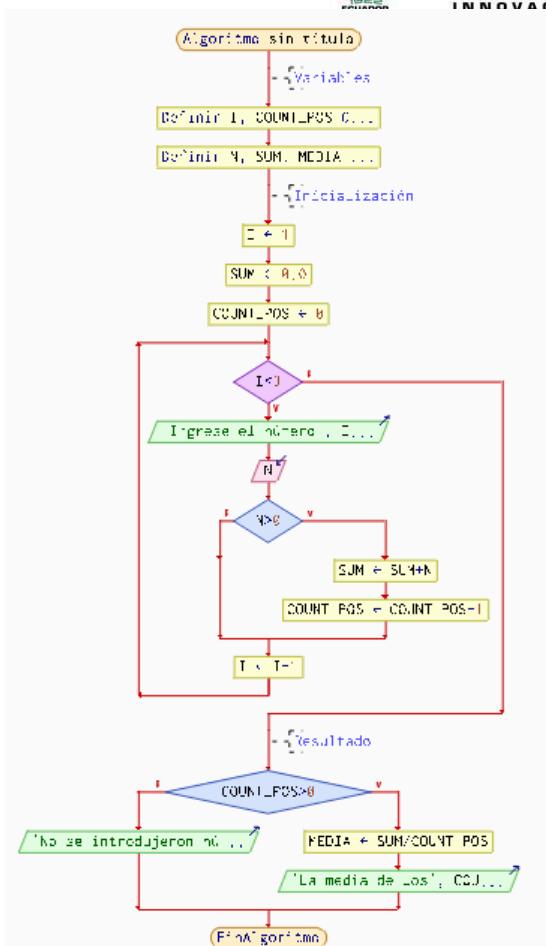
```
principal.c | 26 #incluir <stdio.h>
27
28 int principal () vacío ) {
4     largo largo X ;
5
1     printf ( " Ingrese el número entero de segundos: " ) ;
2     si ( scanf ( "%lld" , & X ) != 1 ) {
3         fprintf ( stderr , " Entrada inválida \n " ) ;
4         devuelve 1 ;
5     }
6
aporte
Ingrese el número entero de segundos: 3600
3600 segundos equivalen a: 0 días, 1 horas, 0 minutos y 0 segundos

...Program finished with exit code 0
Press ENTER to exit console.
```

<https://onlinegdb.com/FFjZs41No>

Desarrolle un programa que lea tres números reales desde teclado y escribe por pantalla la media aritmética únicamente de los números positivos leidos. Puede dar dos opciones A9 Empleando una iteración con salida en cabeza y B) Empleando una iteración con salida en cola

Variable	Nombre	Valor	Tipo
Valor	A	Variable	Real
Valor	B	Variable	Real
Valor	C	Variable	Real
media	media	Variable	Real



principal.c

```
1 if ( opcion == 'A' || opcion == 'a' ) {
2     version_cabeza ( );
3 } else if ( opción == 'B' || opción == 'b' ) {
4     versión_colas ( );
5 } demás {
6     printf ( " Opción no válida. Use A o B. \n " );
7 }
8 devuelve 0 ;
```

```
aporte
alcular la media de los números positivos (entre 3 leídos)
 Elija versión: A) iteración con salida en cabeza (while)
                  B) iteración con salida en cola (do-while)
opción (A/B): a
ingrese el número 1: 4
ingrese el número 2: 2
ingrese el número 3: 6
la media de los 3 números positivos es: 4.000000

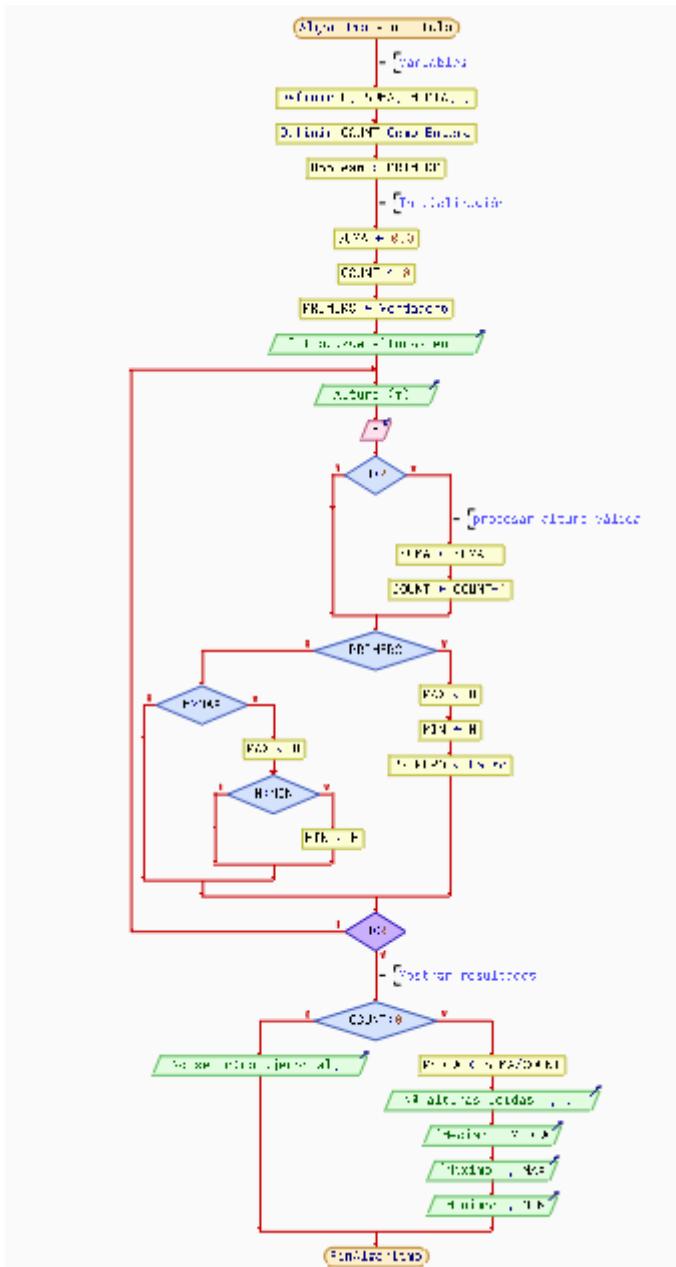
..Program finished with exit code 0
Press ENTER to exit console.
```

Ejercicio 7 Máximo y media de alturas

Desarrolle un programa que lea por teclado un número indeterminado de alturas y que deje de leer

cuando se hay introducido una haltura negativa. Elprograma debe calcular e imprimir la media, el máximo uy el mínimo de las alturas introducidas.

Variable	Nombre	Valor	Tipo
Valor 1	Altura	Variable	Numero
Valor 2	Altura	Variable	Numero
Valor 3	Altura	Variable	Numero
Media	Media	Variable	Numero
Máximo	Máximo	Variable	Numero
Mínimo	Mínimo	Variable	Numero



```

9 #incluir <stdio.h>
10 |
11 void versión_bandera ( void ) {
12     doble H , suma = 0.0 , máx = 0.0 , mín = 0.0 ;
13     int conteo = 0 ;
14     int primero = 1 ; /* bandera */
15
16     printf ( " Versión A) Con bandera PRIMERO \n " );
104    printf ( " Introduzca alturas en metros (valor negativo para terminar): \n " )
105
106    mientras ( 1 ) {
107        printf ( " Altura (m): " );

```

aporte

```

Estadística de alturas. Elija versión:
A) Iteración con salida en cabeza (bandera PRIMERO)
B) Iteración con salida en cola (comprobando COUNT)
Opción (A/B): a
Version A) Con bandera PRIMERO
Introduzca alturas en metros (valor negativo para terminar):
Altura (m): 12
Altura (m): 25
Altura (m): 30
Altura (m): -5

Nº alturas leídas: 3
Media: 22.333333
Máximo: 30.000000
Mínimo: 12.000000

...Program finished with exit code 0
Press ENTER to exit console.

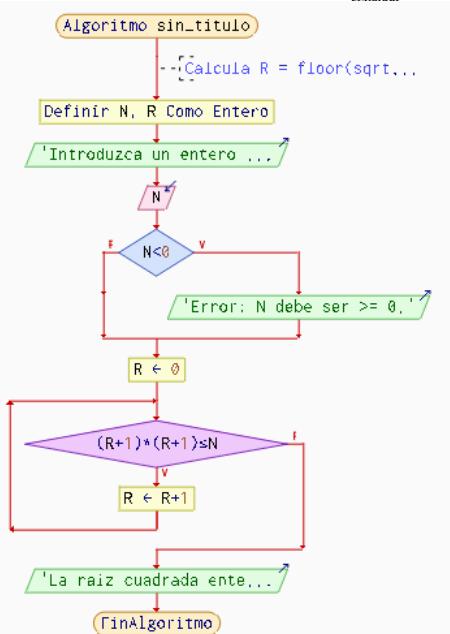
```

<https://onlinegdb.com/hNLHlaZEj>

Ejercicio 8 Raiz Cuadrada Entera

Desarrollo un programa que calculi la raiz cuadrada entera por defecto de un número N positive dado, que se define como el mayor número entero R>0, tal que R Elevado al cuadrado sea $\leq N$.

Variable	Nombre	Valor	Tipo
Valor	N	Variable	Entero
Valor	R	Variable	Entero



```

15 #incluir <stdio.h>
16
104 int principal ( vacío ) {
105     N larga sin signo ;
106     unsigned long long R = 0 ;
107
108     printf ( "Ingrese un entero N (N >= 0): " );
109     si ( scanf ( "%llu" , & N ) != 1 ) {
110         fprintf ( stderr , "Entrada inválida. \n" );
111         devuelve 1 ;
112     }
113
114     mientras ( ( R + 1 ) <= N / ( R + 1 ) ) {
115         R++;
116     }
117
118     devuelve R;
119 }

```

Introduzca un entero N (N >= 0): 4

La raiz cuadrada entera por defecto de 4 es: 2

...Program finished with exit code 0

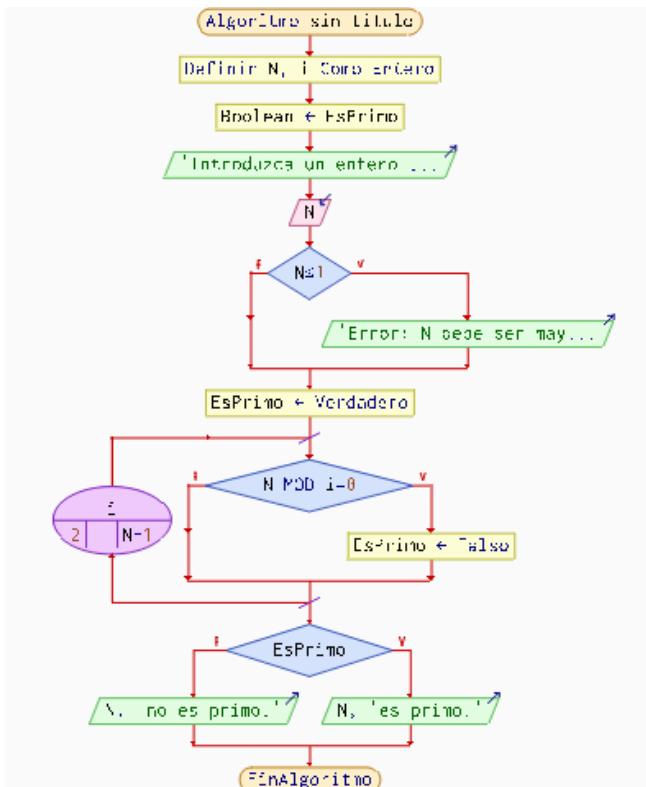
Press ENTER to exit console.

<https://onlinegdb.com/eJxa6aeUk>

Ejercicio 9 Número primo

Desarrolle un programa que lea por teclado un número entero mayor que 1 y determine si es primo o no

Variable	Nombre	Valor	Tipo
Verdadero	Primo	Variable	Entero
Valor	Número	Variable	Entero
Verdadero	Prima	Variable	Entero



```

15         esPrimo = 0 ;
108     romper ; /* ya hemos encontrado un divisor */
15
109 }
16
1   si ( esPrimo )
3   printf ( "%lld es primo. \n " , N ) ;
2 demás
4   printf ( "%lld no es primo (divisible por %d ). \n " , N , i ) ;
106
107 devuelve 0 ;
16
  
```

Introduzca un entero N (N > 1): 5
5 es primo.

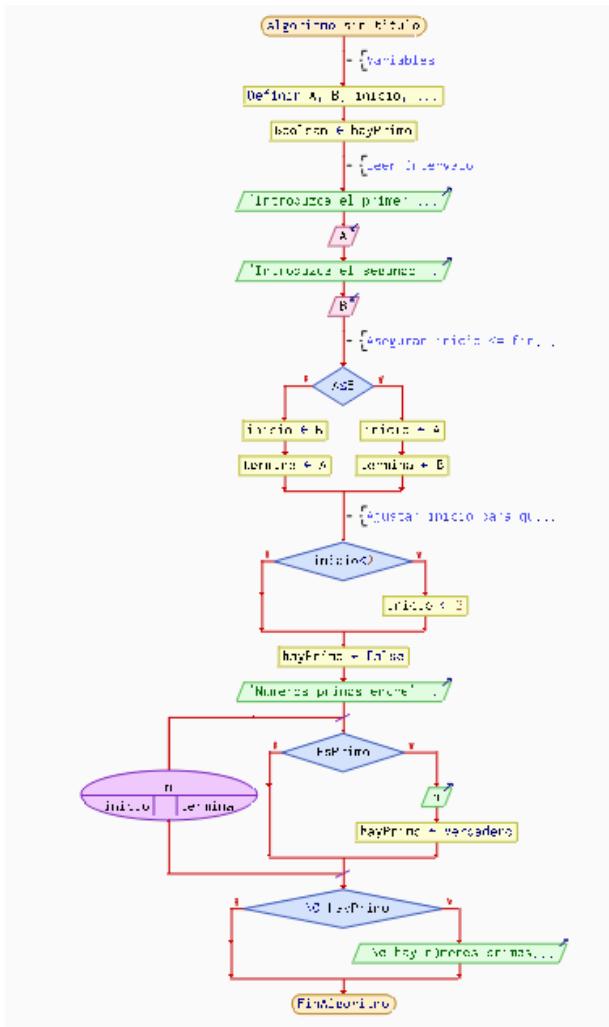
...Program finished with exit code 0
Press ENTER to exit console. □

https://onlinegdb.com/1TKMuSy_E

Ejercicio 10 Números primos comprendidos entre dos números dados

Desarrolle un programa que muestre por pantalla todos los números primos comprendidos entre dos números enteros introducidos por teclado.

Variable	Nombre	Valor	Tipo
Valor	inicial	Variable	Entero
Valor	Termino	Variable	Entero
valor	EsPrimo	Variable	Entero



```

15 #incluir <stdio.h>
109 #include <math.h>
16
17 int esPrimo ( largo largo n ) {
18     si ( n < 2 ) devuelve 0 ;
19     si ( n == 2 ) devuelve 1 ;
20     si ( n % 2 == 0 ) devuelve 0 ;
106
107     largo largo límite = ( largo largo ) sqrt (( doble ) n ) ;
16    para ( largo largo i = 3 ; i <= límite ; i += 2 ) {
29        si ( n % i == 0 ) devuelve 0 ;
28    }
27    devuelve 1 ;

```

▼ ↻ ⌂ ⚙ 🎯 aporte

```

Introduzca el primer entero: 4
Introduzca el segundo entero: 6
Primos entre 4 y 6:
5

...Program finished with exit code 0
Press ENTER to exit console. █

```

<https://onlinegdb.com/STKXoT6Hp>

Enunciado del ejercicio

Desarrolle un programa con un MENÚ PRINCIPAL que ofrezca las siguientes opciones:

MENÚ PRINCIPAL

1. Serie Fibonacci
2. Operaciones
 - 2.1 Par o Impar
 - 2.2 Primo
 - 2.3 Retornar
3. Salir

Requisitos:

- Utilizar estructuras repetitivas (Mientras, Repetir, Para) según corresponda.
- Implementar un submenu dentro de la opción 2 para ejecutar operaciones numéricas.
- Validar las entradas del usuario (solo números positivos).
- Mostrar mensajes claros de retorno y finalización del programa.

Prueba de escritorio (para prueba)

Entrada	Opción seleccionada	Salida esperada
---------	---------------------	-----------------

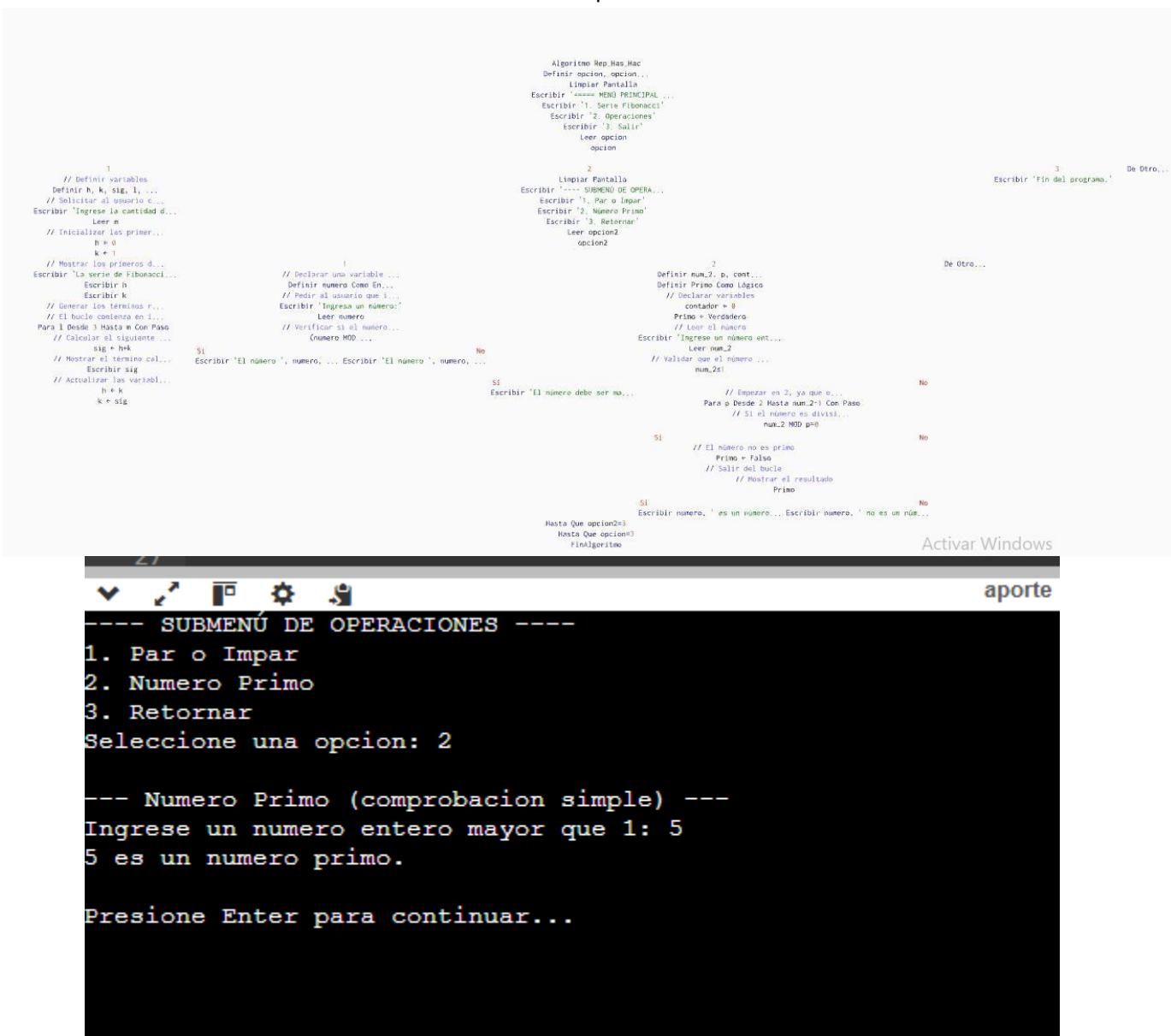


1 y n=5	Fibonacci	0 1 1 2 3
2 → 1 y num=6	Par/Impar	6 es PAR
2 → 2 y num=7	Primo	7 es PRIMO
3	Salir	Fin del programa

Diagrama de flujo (referencial)

El diagrama debe mostrar:

1. Un bloque de decisión principal con tres opciones.
 2. Un subdiagrama para el submenú de operaciones (dos procesos y un retorno).
 3. Flechas de control hacia la condición del ciclo Repetir...Hasta.



```
---- SUBMENÚ DE OPERACIONES ----
1. Par o Impar
2. Numero Primo
3. Retornar
Seleccione una opcion: 1

--- Par o Impar ---
Ingresa un numero entero: 4
El numero 4 es par.

Presione Enter para continuar...■
```