Pontificia Universidad Católica del Perú

Estudios Generales Ciencias

1INF01 - Fundamentos de Programación

Guía de laboratorio #2
Elaboración de programas con estructuras algorítmicas selectivas

4 de septiembre de 2022

Índice general

Historial de Revisiones

Revisión	**Fecha**	**Autor(es)**	**Descripción**
1.0	02.09.2018	A. Melgar	Versión inicial.
1.1	30.03.2019	A. Melgar	Se incrementó la cantidad de problemas propuestos, se añadió color al código en lenguaje C y se completaron los casos de prueba de los problemas propuestos.
1.1	15.04.2019	L. Hirsh	Revisión de la versión 1.1.
2.0	06.07.2019	A. Melgar	Se cambio el contenido del documento para incluir tanto la estructura algorítmica selectiva simple como la estructura algorítmica selectiva doble.
2.0	21.08.2019	C. Aguilera	Revisión de la versión 2.0.
2.0	24.08.2019	A. Melgar	Corrección de la revisión de la versión 2.0 realizada por el profesor César Aguilera.
2.1	17.09.2020	S. Vargas	Clasificación de los ejercicios propuestos en nivel básico, intermedio y avanzado. Se agregaron ejercicios.
2.2	09.09.2021	D. Allasi	Se agregaron ejercicios.
3.0	28.03.2022	D. Allasi, S.Vargas	Se cambio el contenido del documento para incluir la estructura selectiva anidada. Se agregaron ejercicios.

Siglas

EEGGCC	Estudios Generales Ciencias
IDE	Entorno de Desarrollo Integrado
PUCP	Pontificia Universidad Católica del Perú

Capítulo 1

Guía de Laboratorio #2

1.1. Introducción

Esta guía ha sido diseñada para que sirva como una herramienta de aprendizaje y práctica para el curso de Fundamentos de Programación de los Estudios Generales Ciencias (EEGGCC) en la Pontificia Universidad Católica del Perú (PUCP). En particular se focaliza en el tema "Elaboración de programas con estructuras algorítmicas selectivas".

Se busca que el alumno resuelva paso a paso las indicaciones dadas en esta guía contribuyendo de esta manera a los objetivos de aprendizaje del curso, en particular en el diseño de programas con estructuras algorítmicas selectivas usando el paradigma imperativo. Al finalizar el desarrollo de esta guía y complementando lo que se realizará en el correspondiente laboratorio, se espera que el alumno:

- Comprenda el control de flujo de un programa en el paradigma imperativo y en particular la estructura algorítmica selectiva.
- Diseñe algoritmos expresados en diagramas de flujo y pseudocódigos que controlen el flujo usando una estructura algorítmica selectiva.
- Implemente programas que utilicen la estructura algorítmica selectiva en un lenguaje de programación imperativo.
- Implemente programas que utilicen expresiones complejas que combinen operadores relacionales y lógicos.

1.2. Materiales y métodos

Como herramienta para el diseño de pseudocódigos y diagramas de flujo se utilizará PSeInt[1]. El PSeInt deberá estar configurado usando el perfil PUCP definido por los profesores del curso. Como lenguaje de programación imperativo se utilizará el lenguaje C. Como Entorno de Desarrollo Integrado (IDE) para el lenguaje C se utilizará Dev C++[2]. No obstante, es posible utilizar otros IDEs como Netbeans y Eclipse.

1.3. Estructura Algorítmica Selectiva

Los programas escritos en el paradigma imperativo se basan en el cambio de estado de las variables definidas en los programas. Todo programa sigue un flujo, el cual puede ser modificado a partir de estructuras de control

[1] http://pseint.sourceforge.net/
[2] http://sourceforge.net/projects/orwelldevcpp

de flujo. Las estructuras de control de flujo pueden ser de dos tipos: estructuras algorítmicas selectivas y estructuras algorítmicas iterativas.

Las estructuras selectivas, permiten que los programas ejecuten un conjunto de instrucciones si es que cumplen una determinada condición. Es decir, el conjunto de instrucciones se ejecuta una sola vez si cumple la condición. Por otra parte, las estructuras iterativas permiten que los programas ejecuten un conjunto de instrucciones tantas veces como sea necesario, dependiendo de una determinada condición. Las estructuras algorítmicas selectivas por su lado se pueden clasificar en estructuras selectivas simples y estructuras selectivas dobles.

La estructura selectiva simple permite ejecutar un conjunto de instrucciones si y solo si se cumple una determinada condición. Si la condición no se cumple, el conjunto de instrucciones no se ejecuta. La estructura selectiva simple es muy usada en la programación imperativa, gracias a ella se pueden verificar situaciones antes de realizar determinado procesamiento. Muchas situaciones requieren que se verifiquen condiciones iniciales antes de realizar el procesamiento de datos, por ejemplo, para calcular la raíces de una ecuación cuadrática de una variable, si se desea obtener una solución real, el discriminante debe ser mayor o igual a cero. Si se desea calcular el área de un triángulo dados sus 3 lados, antes se debe verificar si efectivamente dichos lados forman un triángulo.

La estructura selectiva doble complementa a la estructura selectiva simple. Al igual que la selectiva simple, la estructura selectiva doble permite ejecutar un conjunto de instrucciones si y solo si se cumple una determinada condición. Si la condición no se cumple, se ejecuta otro conjunto de instrucciones. Esto último es lo que diferencia a la selectiva doble de la selectiva simple. En la selectiva doble se cuentan con dos conjuntos de instrucciones que, dependiendo de la condición, se ejecuta uno o se ejecuta el otro. De ahí el nombre de selectiva doble.

La estructura selectiva doble es muy usada en la programación imperativa, gracias a ella se pueden tomar decisiones sobre cómo realizar determinado procesamiento. Muchas situaciones requieren que se realice el cálculo de diferente manera dependiendo de la condición. Por ejemplo, para calcular las raíces de una ecuación cuadrática de una variable, si el discriminante es mayor o igual a cero, se calcula de una forma pues la solución será real, pero si el discriminante es menor que cero, la solución involucrará un número complejo y el procesamiento, en este caso, es diferente. Otro uso muy común es la utilización de la selectiva doble en conjunción con la selectiva simple para ejecutar diferentes acciones en un menú de opciones; como es por ejemplo el caso de una calculadora, que dependiendo de la operación seleccionada $(+,-,*,/)$, ejecutará diferentes cálculos ($a+b$, $a-b$, $a*b$, a/b).

1.3.1. Representación de la Estructura Algorítmica Selectiva Simple

A continuación, se revisará cómo se representa la estructura selectiva simple tanto en pseudocódigo como en diagrama de flujo, así como su implementación en lenguaje C.

Representación en pseudocódigo

En la figura 1.1 se puede apreciar la representación de la estructura selectiva simple en pseudocódigo. La estructura inicia con el identificador `Si` y finaliza con el identificador `Fin Si`. La `condición` es una expresión lógica como por ejemplo: $i < 100$, $nota \leq 10$, $i \leq max\ y\ i <> 0$. Luego de la `condición` se coloca el identificador `Entonces`. En el `conjunto de instrucciones` se pueden colocar asignaciones ($i \leftarrow 0$), cálculo de expresiones matemáticas ($suma \leftarrow suma + termino$) e inclusive otras estructuras selectivas (`Si i=0 Entonces`).

Si *condición* **Entonces**
| conjunto de instrucciones;
Fin Si

Figura 1.1: Pseudocódigo: Estructura selectiva simple

Representación en diagrama de flujo

En la figura 1.2 se puede apreciar la representación de la estructura selectiva simple en diagrama de flujo. La estructura inicia con el bloque `condición` y si la condición es verdadera, se ejecuta el `conjunto de instrucciones`. En el diagrama, el control del flujo se gestiona a través de las líneas que conectan los bloques. Como se aprecia, inmediatamente después de ejecutar el bloque de instrucciones, el control se dirige hacia abajo. Si la `condición` no se satisface, el flujo se dirige hacia abajo sin procesar el `conjunto de instrucciones`.

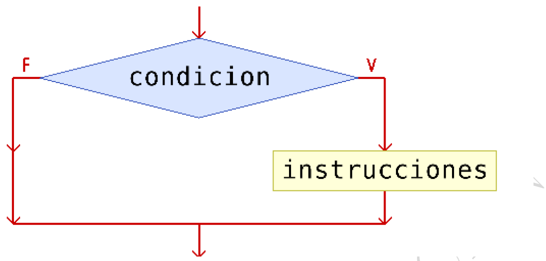

Figura 1.2: Diagrama de Flujo: Selectiva simple

Implementación en lenguaje C

En el lenguaje C la estructura selectiva simple se implementa a través de la instrucción `if`. La representación del `if` se puede apreciar en el programa 1.1. El funcionamiento de la instrucción `if` es muy similar al `Si` del pseudocódigo y diagrama de flujo, pero hay que tener ciertas consideraciones para su uso.

Como se sabe, el lenguaje C no implementa nativamente el tipo de dato lógico (`bool` o `boolean`). Entonces ¿cómo hace el lenguaje C para evaluar la condición de la instrucción `if`? el lenguaje C asume que todo valor igual a 0 falla una condición. El comportamiento es muy similar al $falso$ de una expresión lógica. Por lo contrario, un valor diferente a 0 hará que se cumpla la condición. El comportamiento es muy similar al *verdadero* de una expresión lógica. Por ejemplo: si se tienen las siguientes definiciones `int suma=0, i=10;`, las siguientes expresiones serán consideradas *verdaderas*: $suma <= 100$, $i == 10$, $suma < i$ e i. Por otro lado, las siguientes expresiones serán consideradas $falsas$: $suma >= 100$, $i == 20$, $suma > i$ y $suma$.

Programa 1.1: Lenguaje C: Estructura selectiva simple

```
...
if (condición){
        conjunto de instrucciones;
}
...
```

1.3.2. Representación de la Estructura Algorítmica Selectiva Doble

A continuación, se revisará cómo se representa la estructura selectiva doble tanto en pseudocódigo como en diagrama de flujo, así como su implementación en lenguaje C.

Representación en pseudocódigo

En la figura 1.3 se puede apreciar la representación de la estructura selectiva doble en pseudocódigo. La estructura inicia con el identificador `Si` y finaliza con el identificador `Fin Si`. La `condición` es una expresión lógica como por ejemplo: $i < 100$, $nota <= 10$, $i <= max\ y\ i <> 0$. Luego de la `condición` se coloca el identificador `Entonces`.

Como se mencionó anteriormente, a la estructura selectiva doble se le asocian dos conjuntos de instrucciones, un conjunto de instrucciones que se ejecuta si se cumple la condición (conjunto de instrucciones *a* en la figura

1.3) y otro conjunto de instrucciones sino se cumple la condición (conjunto de instrucciones b en la figura 1.3). Para separar ambos conjuntos de instrucciones se utiliza el identificador `SiNo`.

En ambos `conjuntos de instrucciones` se pueden colocar asignaciones ($i \leftarrow 0$), cálculo de expresiones matemáticas ($suma \leftarrow suma + termino$) e inclusive otras estructuras selectivas (`Si i=0 Entonces`).

Si *condición* **Entonces**
  conjunto de instrucciones a;
SiNo
  conjunto de instrucciones b;
Fin Si

Figura 1.3: Pseudocódigo: Estructura selectiva doble

Representación en diagrama de flujo

En la figura 1.4 se puede apreciar la representación de la estructura selectiva doble en diagrama de flujo. La estructura inicia con el bloque `condición` y si la condición es verdadera, se ejecuta el `conjunto de instrucciones a`. Si la condición es falsa, se ejecuta el `conjunto de instrucciones b`.

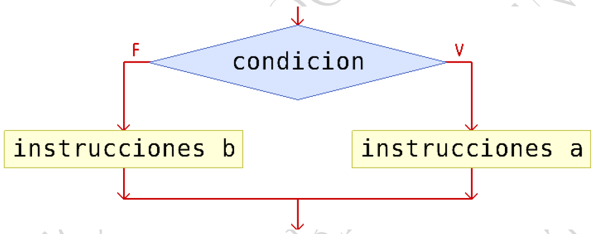

Figura 1.4: Diagrama de Flujo: Selectiva doble

Implementación en lenguaje C

En el lenguaje C la estructura selectiva doble se implementa a través de la instrucción `if`. La representación del `if` se puede apreciar en el programa 1.2. El segundo conjunto de instrucciones se coloca luego del identificador `else`. El conjunto de instrucciones en lenguaje C se delimita por los símbolos `{` y `}`. Esta delimitación es opcional cuando el conjunto de instrucciones está formada por una sola instrucción.

Programa 1.2: Lenguaje C: Estructura selectiva doble

```
...
if (condición){
        conjunto de instrucciones a;
}
else{
        conjunto de instrucciones b;
}
...
```

1.4. Cálculo de las raíces de ecuaciones cuadráticas con una variable - Selectivas simples y dobles

1.4.1. Verificación del discriminante

Una ecuación cuadrática con una variable es una ecuación que tiene la forma de $ax^2 + bx + c = 0$ siendo a, b y c números reales con la restricción que $a \neq 0$. Para encontrar la solución a la ecuación se puede utilizar la siguiente fórmula general $x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$. La fórmula general produce una solución con dos raíces, la $x_1 = \frac{-b+\sqrt{b^2-4ac}}{2a}$ y la $x_2 = \frac{-b-\sqrt{b^2-4ac}}{2a}$ las cuales no son necesariamente diferentes.

Dada una ecuación cuadrática con una variable se solicita que elabore un algoritmo expresado en diagrama de flujo y pseudocódigo así como un programa en lenguaje C que calcule las 2 raíces de la solución. En la guía #1 se resolvió este problema y se asumió que el discriminante siempre sería mayor o igual a 0 por lo que la solución daría siempre un número real. En la realidad no se puede asumir esto pues el usuario puede ingresar los números que desee. ¿Qué se debe hacer entonces? Antes de procesar se debe verificar si los datos ingresados permiten dicho procesamiento. En este caso en particular, primero se debe hacer la lectura de los datos de la ecuación, luego calcular el discriminante y posteriormente verificar si este es mayor o igual a 0. Esta verificación se realiza con una estructura selectiva simple. Solo si se cumple la condición ($discriminante \geq 0$), se procede al cálculo de las raíces. Si no se cumple la condición, no se hace nada.

En la figura 1.5 se puede apreciar el diagrama de flujo que diseña la alternativa de solución propuesta. Note la inclusión de la estructura selectiva simple (bloque con la condición $discriminante >= 0$).

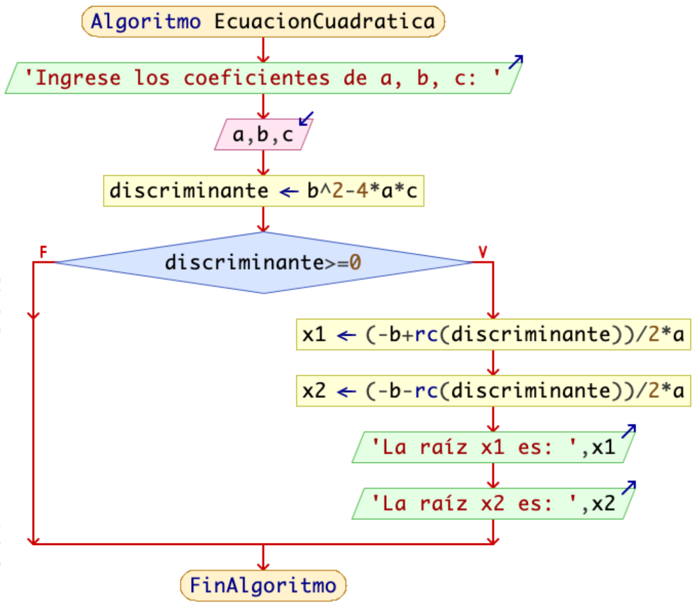

Figura 1.5: Diagrama de flujo: Ecuación cuadrática con estructura algorítmica selectiva simple

El problema con el diagrama de flujo que se visualiza en la figura 1.5 es que si no hay solución real ($discriminante < 0$), el algoritmo no imprime nada. Para solucionar este problema se puede utilizar una estructura algorítmica selectiva doble que permita evaluar este caso en particular, de forma tal que cuando no se pueda retornar una solución real, se presente el mensaje informativo adecuado. La alternativa de solución con la estructura algorítmica selectiva doble se puede apreciar en la figura 1.6.

En la figura 1.7 se aprecia el pseudocódigo de la última alternativa de solución. La estructura selectiva doble inicia en la línea 5 y finaliza en la línea 12. Por razones estéticas, los conjuntos de instrucciones de la estructura selectiva doble se indenta, es decir se crea un sangrado. Todo el bloque se mueve unos caracteres a la derecha. Si bien es cierto a las herramientas que compilan el código no les afecta la indentación de código, a los humanos

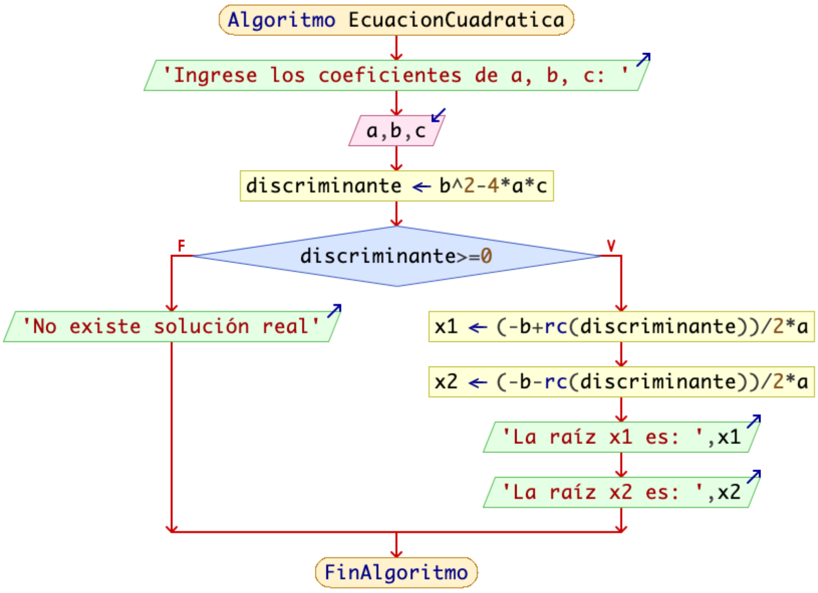

Figura 1.6: Diagrama de flujo: Ecuación cuadrática con estructura algorítmica selectiva doble

sí. Es mucho más fácil entender un código bien indentado que un código sin indentación. Esto afecta mucho la etapa de corrección de errores y al mantenimiento de software.

```
Algoritmo EcuacionCuadratica
    Escribir 'Ingrese los coeficientes de a, b, c: '
    Leer a,b,c
    discriminante <- b^2-4*a*c
    Si discriminante>=0 Entonces
        x1 <- (-b+rc(discriminante))/2*a
        x2 <- (-b-rc(discriminante))/2*a
        Escribir 'La raíz x1 es: ',x1
        Escribir 'La raíz x2 es: ',x2
    SiNo
        Escribir 'No existe solución real'
    FinSi
FinAlgoritmo
```

Figura 1.7: Pseudocódigo: Ecuación cuadrática

En el programa 1.3 se puede apreciar la alternativa de solución en lenguaje C. La estructura selectiva doble inicia en la línea 12 y finaliza en la línea 19. Al igual que en el pseudocódigo, el conjunto de instrucciones de la selectiva doble se encuentra indentado. Un aspecto importante de la sintaxis del lenguaje C es que cuando el conjunto de instrucciones está conformado por una sola instrucción, no es necesario delimitar el bloque con los símbolos { y }. Esto se puede apreciar en el bloque del `else`, entre las líneas 18 y 19. Otro detalle importante en lenguaje C es que la expresión que representa a la condición en la estructura selectiva doble siempre va entre paréntesis. Los paréntesis no son necesarios para el caso del diagrama de flujo ni para el pseudocódigo en PSeInt.

Programa 1.3: Verificación de discriminante

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main() {
```

```c
    int a, b, c;
    double discriminante, x1, x2;

    printf("Ingrese coeficientes a, b y c de la ecuación: ");
    scanf(" %d %d %d", &a, &b, &c);

    discriminante = pow(b, 2) - 4 * a*c;
    if (discriminante >= 0) {
        x1 = (-b + sqrt(discriminante)) / 2 * a;
        x2 = (-b - sqrt(discriminante)) / 2 * a;
        printf("La raíz x1 es %lf\n", x1);
        printf("La raíz x2 es %lf\n", x2);
    }
    else
        printf("No existe solución real");
    return 0;
}
```

Para poner en práctica

¿Cuáles de las siguientes ecuaciones tiene solución real?

- $x^2 + x + 1 = 0$
- $x^2 + 3x + 6 = 0$
- $2x^2 + 3x + 4 = 0$
- $-2x^2 + 3x + 4 = 0$
- $-x^2 - 2x + 4 = 0$
- $x^2 - 4x - 7 = 0$

1.4.2. Cálculo de las raíces complejas

En la guía #1 se realizó el cálculo de las raíces de una ecuación cuadrática con una variable y se asumió que el discriminante siempre sería mayor o igual a 0 por lo que la solución daría siempre un número real. En la sección anterior se ha planteado una solución en donde se ha verificado que el discriminante sea siempre mayor o igual a 0 para evitar errores de procesamiento. Pero en el mundo de las matemáticas, si el discriminante es menor que 0, existe una solución compleja. ¿Cómo se puede adaptar el programa para que calcule una raíz real si es que el discriminante es mayor o igual a 0 y calcule una raíz compleja si el discriminantes es menor que 0? Este es una situación ideal para una estructura selectiva doble.

Se puede diseñar una estructura selectiva doble para que cuando el discriminantes sea mayor o igual a 0, siga la misma solución ya analizada en la sección anterior. Sino se cumple la condición aparece un conjunto de instrucciones diferente, uno que permita calcular las raíces como números complejos. Este nuevo conjunto de instrucciones es similar al anterior, solo que se debe obtener el valor absoluto del discriminante para poder calcular la raíz cuadrada. En PSeInt el valor absoluto se puede obtener mediante la función `abs`.

Recordar que:

En el contexto de los números complejos $i = \sqrt{-1}$

En la figura 1.8 se puede apreciar el diagrama de flujo que diseña la alternativa de solución propuesta. Note la inclusión de la estructura selectiva doble (bloque con la condición *discriminante* $>= 0$).

En la figura 1.9 se aprecia el pseudocódigo de la misma alternativa de solución. La estructura doble simple inicia en la línea 5 y finaliza en la línea 16. El bloque de instrucciones que se ejecuta cuando la condición se cumple se encuentra entre las líneas $6 - 9$. El bloque de instrucciones que se ejecuta cuando la condición no se cumple se encuentra entre las líneas $11 - 15$. Por razones estéticas, los conjuntos de instrucciones de la estructura selectiva doble se indenta.

En el programa 1.4 se puede apreciar la alternativa de solución en lenguaje C. La estructura selectiva doble inicia en la línea 12 y finaliza en la línea 26. El bloque de instrucciones que se ejecuta cuando la condición se cumple se encuentra entre las líneas $13 - 17$. El bloque de instrucciones que se ejecuta cuando la condición no

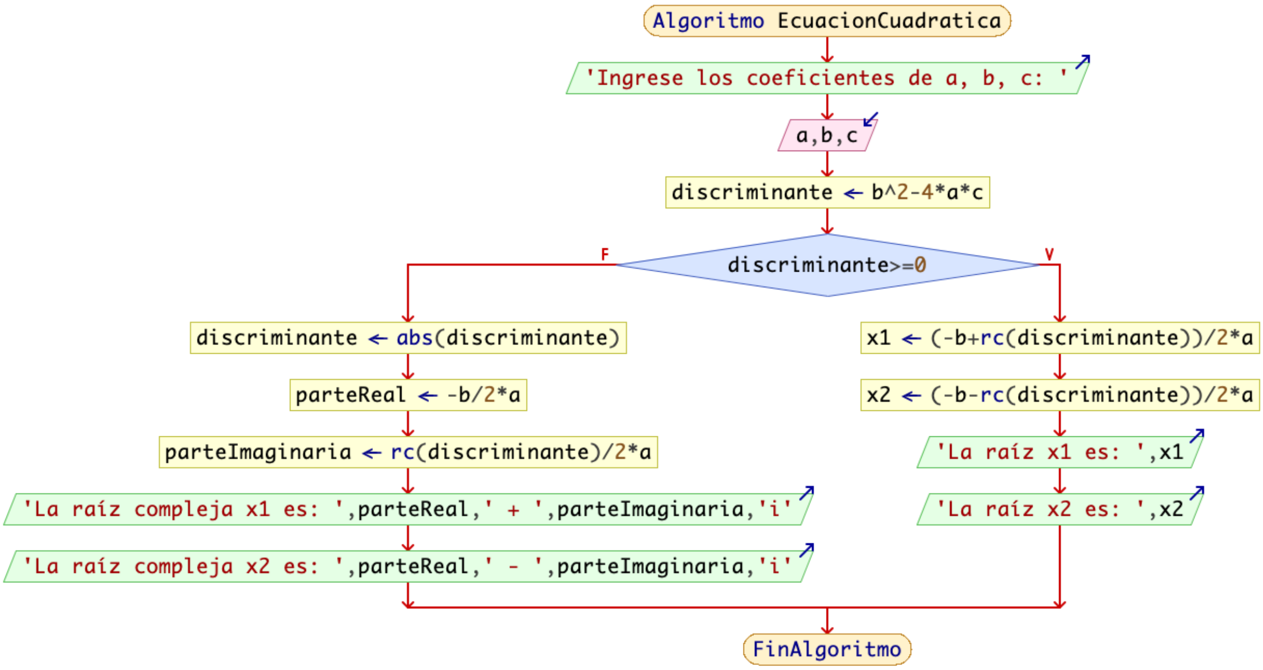

Figura 1.8: Diagrama de flujo: Ecuación cuadrática con raíces reales y complejas

```
Algoritmo EcuacionCuadratica
    Escribir 'Ingrese los coeficientes de a, b, c: '
    Leer a,b,c
    discriminante <- b^2-4*a*c
    Si discriminante>=0 Entonces
        x1 <- (-b+rc(discriminante))/2*a
        x2 <- (-b-rc(discriminante))/2*a
        Escribir 'La raíz x1 es: ',x1
        Escribir 'La raíz x2 es: ',x2
    SiNo
        discriminante <- abs(discriminante)
        parteReal <- -b/2*a
        parteImaginaria <- rc(discriminante)/2*a
        Escribir 'La raíz compleja x1 es: ',parteReal,' + ',parteImaginaria,'i'
        Escribir 'La raíz compleja x2 es: ',parteReal,' - ',parteImaginaria,'i'
    FinSi
FinAlgoritmo
```

Figura 1.9: Pseudocódigo: Ecuación cuadrática con raíces reales y complejas

se cumple se encuentra entre las líneas $20-25$. Al igual que en el pseudocódigo, los conjuntos de instrucciones de la selectiva doble se encuentran indentados.

Hay dos aspectos importantes a comentar en relación al programa 1.4. El primero viene en relación a la declaración de las variables `x1`, `x2`, `parteReal` y `parteImaginaria`. Como usted podrá notar en la línea 13 del programa 1.4 se declaran las variables `x1` y `x2` y en la línea 20 del mismo programa, se declaran las variables `parteReal` y `parteImaginaria`. El lenguaje C permite que las variables se puedan declarar dentro de cada bloque de instrucciones. Lo único que hay que tener en consideración es que dicha variable no se podrá usar fuera del bloque en donde se declara. Esto es muy útil cuando se quiere encapsular el procesamiento a ciertos conjuntos de instrucciones.

El otro aspecto a considerar es el cálculo del valor absoluto en lenguaje C. Como puede notar, en la línea 21 del programa 1.3 se utiliza la función `fabs`. Esta función permite calcular el valor absoluto de un número real. Se encuentra declarada en el archivo de cabecera `math.h`. Para calcular el valor absoluto de un número entero se utiliza la función `abs` que se encuentra declarada en el archivo de cabecera `stdlib.h`.

Programa 1.4: Ecuación cuadrática con raíces reales y complejas

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

int main() {
    int a, b, c;
    double discriminante;

    printf("Ingrese coeficientes a, b y c de la ecuación: ");
    scanf(" %d %d %d", &a, &b, &c);

    discriminante = pow(b, 2) - 4 * a*c;
    if (discriminante >= 0) {
        double x1, x2;
        x1 = (-b + sqrt(discriminante)) / 2 * a;
        x2 = (-b - sqrt(discriminante)) / 2 * a;
        printf("La raíz real x1 es %lf\n", x1);
        printf("La raíz real x2 es %lf\n", x2);
    }
    else{
        double parteReal, parteImaginaria;
        discriminante = fabs(discriminante);
        parteReal = -b/2*a;
        parteImaginaria = sqrt(discriminante)/2*a;
        printf("La raíz compleja x1 es %.2lf + %.2lfi\n", parteReal, parteImaginaria);
        printf("La raíz compleja x2 es %.2lf - %.2lfi\n", parteReal, parteImaginaria);
    }
    return 0;
}
```

A continuación, sigue un ejemplo de ejecución de este programa:

```
Ingrese coeficientes a, b y c de la ecuación: 1 -4 13
La raíz compleja x1 es 2.00 + 3.00i
La raíz compleja x2 es 2.00 - 3.00i
```

Para poner en práctica

¿Cuáles son las raíces de las siguientes ecuaciones?

- $x^2 + x + 1 = 0$
- $x^2 + 3x + 6 = 0$
- $2x^2 + 3x + 4 = 0$
- $-2x^2 + 3x + 4 = 0$
- $-x^2 - 2x + 4 = 0$
- $x^2 - 4x - 7 = 0$

1.5. Verificación de datos de entrada para el problema de cambio de billetes

Un cajero electrónico posee billetes de las siguientes denominaciones 50, 20 y 10. Dada una cantidad x de dinero, se desea conocer la menor cantidad de billetes que se requiere para obtener la cantidad x de dinero. En caso no se consiga la cantidad exacta, deberá indicarse además el monto faltante para llegar a x.

El problema en cuestión fue resuelto en la guía #1, pero en dicha solución se asumió que el usuario siempre ingresaría una cantidad $x > 0$. Nuevamente, el computador no tiene control sobre el usuario y por esto se hace necesaria la realización de una verificación antes del procesamiento de los datos.

En el programa 1.5 se puede observar una alternativa de solución al problema incluyendo la verificación del monto ingresado por el usuario. La verificación se puede apreciar en la línea 10. Si el usuario no ingresa un monto "válido", no ejecutará el procesamiento, en este caso se presentará una mensaje informativo.

Programa 1.5: Verificación del monto de las monedas

```c
1  #include <stdio.h>
2  #include <math.h>
3
4  int main() {
5      int x, cant50, cant20, cant10, resto;
6
7      printf("Ingrese el monto: ");
8      scanf(" %d", &x);
9
10     if (x > 0) {
11         cant50 = x / 50;
12         x %= 50;
13         cant20 = x / 20;
14         x %= 20;
15         cant10 = x / 10;
16         resto = x % 10;
17         printf("Se requieren:\n");
18         printf("\t %d billete de 50\n", cant50);
19         printf("\t %d billete de 20\n", cant20);
20         printf("\t %d billete de 10\n", cant10);
21         printf("\t %d es el monto faltante\n", resto);
22     }
23     else
24         printf("Ingrese un monto mayor que cero.");
25     return 0;
26 }
```

Para poner en práctica

- Pruebe el programa introduciendo valores negativos o iguales a cero y vea qué sucede.
- Diseñe el algoritmo expresado en diagrama de flujo que corresponde al programa 1.5.
- Diseñe el algoritmo expresado en pseudocódigo que corresponde al programa 1.5.

1.6. Verificación de la base y operando en el cálculo de logaritmos

Sea b un número real positivo no nulo distinto de 1, y x otro número positivo no nulo. Se denomina logaritmo del número x en la base b, al exponente l al que debe elevarse la base b para obtener dicho número x. El logaritmo se expresa de la siguiente manera $\log_b x$ y si $\log_b x = l \leftrightarrow b^l = x$. Por ejemplo $\log_5 625 = 4$ ya que $625 = 5^4 = 5 \times 5 \times 5 \times 5$. El logaritmo es la operación inversa a la exponenciación.

Existen diversas propiedades de los logaritmos pero en esta sección centraremos la atención en el teorema de cambio de base. Según este teorema $\log_b x = \frac{\log_a x}{\log_a b}$. Esto significa que se puede basar el cálculo del logaritmo en cualquier base con el logaritmo de otra base. Este teorema es muy importante pues algunos lenguajes de programación solamente ofrecen la función logaritmo en una base determinada, típicamente el logaritmo natural cuya base es el número e. Aplicando el teorema de cambio de base y fijando la base en el número e, se tiene que $\log_b x = \frac{\ln x}{\ln b}$.

Para poder realizar el cálculo del logaritmo de un número x en una base b debe cumplirse que $b > 0 \wedge b \neq 1$ y $x > 0$. ¿Cómo se expresa dicha condición en lenguaje C? En la guía #1 se estudiaron los operadores en lenguaje C, la condición anteriormente descrita se puede expresar como `b>0 && b!=1 && x>0`. En el programa 1.6 se puede apreciar la validación antes de realizar el cálculo del logaritmo. Es bastante común que la expresión que representa a la condición en una estructura algorítmica selectiva (y en realidad en todas las estructuras de control de flujo) contenga expresiones complejas que combinen tanto operadores relacionales como operadores lógicos.

Programa 1.6: Verificación de la base y número de un logaritmo

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

int main() {
    double base, numero, logaritmo;

    printf("Ingrese el número y la base: ");
    scanf(" %lf %lf", &numero, &base);

    if (base > 0 && base != 1 && numero > 0) {
        logaritmo = log(numero) / log(base);
        printf("El logaritmo es %lf\n", logaritmo);
    }
    else
        printf("Los datos ingresados son inconsistentes\n");
    return 0;
}
```

Para poner en práctica

- Diseñe el algoritmo expresado en diagrama de flujo que corresponde al programa 1.6.
- Diseñe el algoritmo expresado en pseudocódigo que corresponde al programa 1.6.

1.7. ¿3 lados forman un triángulo?

Según el teorema de la desigualdad del triángulo *“La suma de las longitudes de cualesquiera de los lados de un triángulo es mayor que la longitud del tercer lado”* (ver figura 1.10).

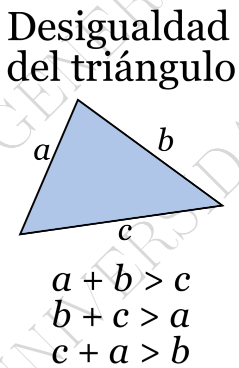

Figura 1.10: Teorema de la desigualdad. Imagen disponible en la URL `https://commons.wikimedia.org/wiki/File:Desigualdad_del_tri%C3%A1ngulo.svg`.

Basándose en el teorema de la desigualdad del triángulo, para verificar si 3 lados a, b y c forman en realidad un triángulo, deben cumplirse 3 condiciones: i) $a + b > c$, ii) $b + c > a$ y iii) $c + a > b$. ¿Cómo expresar estas 3 condiciones en lenguaje C? Esta expresión se representará como una proposición lógica compuesta, usando la conjunción para unir las tres condiciones en la proposición. En lenguaje C el operador de conjunción se representa mediante el símbolo `&&`. Para una mejor lectura de la condición $x + y > z$ se agrupará la suma usando paréntesis de la siguiente manera: `(x+y)>z`. La solución propuesta para este problema se puede apreciar en el programa 1.7.

Programa 1.7: Verificación de los lados de un triángulo

```c
#include <stdio.h>

int main() {
```

```c
    int a, b, c, lados_forman_triangulo;

    printf("Ingrese los lados de un triángulo: ");
    scanf(" %d %d %d", &a, &b, &c);

    lados_forman_triangulo = (a+b)>c && (a+c)>b && (b+c)>a;
    if (lados_forman_triangulo)
        printf("Los lados forman un triángulo.\n");
    else
        printf("Los lados no forman un triángulo.\n");
    return 0;
}
```

Para poner en práctica

- Si cambia la condición de la selectiva por $a+b>c$ && $a+c>b$ && $b+c>a$ ¿El programa sigue funcionando?.
- Diseñe el algoritmo expresado en diagrama de flujo que corresponde al programa 1.7.
- Diseñe el algoritmo expresado en pseudocódigo que corresponde al programa 1.7.

1.8. ¿Es el año bisiesto?

Si queremos entender por qué existen los años bisiestos debemos fijarnos en el movimiento de la Tierra alrededor del Sol: nuestro planeta rota $365,24219$ veces durante una órbita completa alrededor del astro, por tanto un año dura 365 días, 5 horas, 48 minutos y 56 segundos, no 365 días exactos.

Al emperador Julio César se le ocurrió crear el año bisiesto. Si cada año nosotros contamos esos 365 días, perdemos esas 5 horas que deberemos recuperar. Durante tres años contamos esos 365 y al cuarto recuperamos el día que falta, los 29 días que tiene febrero, el año bisiesto.

El año bisiesto tiene una buena explicación. Si no añadiéramos un día completo cada cuatro años, las estaciones acabarían descompasadas del calendario, de tal manera que después de unos 700 años, en el hemisferio norte la Navidad caería en mitad del verano. Al revés, en el hemisferio sur[3].

¿Cómo se puede determinar si un año es bisiesto? Un año es bisiesto si el número que lo representa es divisible entre 4, salvo que sea año secular -último de cada siglo, terminado en 00-, en cuyo caso también ha de ser divisible entre 400.

Si tenemos las siguientes preposiciones:

- p: El número que representa al año es divisible entre 4
- q: El número que representa al año es divisible entre 100
- r: El número que representa al año es divisible entre 400

La expresión lógica que permite determinar si un año es bisiesto es $p \wedge (\neg q \vee r)$. En el programa 1.8 se puede apreciar la implementación de esta expresión lógica en lenguaje C.

Programa 1.8: Verificación de un año bisiesto

```c
#include <stdio.h>

int main() {
    int anho, p, q, r, es_bisiesto;

    printf("Ingrese el año: ");
    scanf(" %d", &anho);
```

[3] Texto tomado del a URL https://elpais.com/elpais/2016/02/29/actualidad/1456703053_016861.html

```

    p = (anho % 4) == 0;
    q = (anho % 100) == 0;
    r = (anho % 400) == 0;
    es_bisiesto = p && (!q || r);

    if (es_bisiesto)
        printf("El año es bisiesto.\n");
    else
        printf("El año no es bisiesto.\n");
    return 0;
}
```

1.9. Estructuras Algorítmicas Selectivas Anidadas

Los algoritmos siguen un flujo de ejecución el cual puede ser modificado. La limitación que trae la estructura algorítmica selectiva, es que a lo más se pueden tomar decisiones sobre dos alternativas de flujo. ¿Cómo se podría tomar decisiones sobre más de dos alternativas de flujo? Una alternativa de respuesta a esta pregunta es la utilización de la estructura algorítmica selectiva doble en cuyo interior se incluya otra estructura algorítmica selectiva. A esto se le conoce como anidación de estructuras algorítmicas selectivas y son muy utilizadas cuando un algoritmo requiere evaluar diversas alternativas.

Las estructuras algorítmicas selectivas son muy usadas en la programación imperativa pues las decisiones que se deben tomar en los problemas reales incluyen por lo general varias alternativas. Por ejemplo para calcular la raíces de una ecuación cuadrática de una variable, si el discriminante es mayor a cero, existen dos raíces reales, si el discriminantes es igual a cero, existe una sola raíz real y si el discriminante es menor que cero, existen dos raíces complejas. Es decir, existen varias alternativas para calcular las raíces de una ecuación cuadrática que dependen de el valor que tome el discriminante.

1.9.1. Representación de la Estructura Selectiva Anidada

En realidad la estructura selectiva anidada no es una nueva estructura sino la utilización de la misma estructura de forma recurrente tantas veces como sea necesario. A continuación se presentará la anidación de estructuras selectivas tanto en pseudocódigo como en diagrama de flujo, así como en el lenguaje C.

1.9.2. Representación en pseudocódigo

En la figura 1.11 se puede apreciar la anidación de dos estructuras algorítmica selectivas dobles. Existen dos condiciones que permiten que se pueda decidir entre 3 conjuntos de instrucciones. La primera estructura algorítmica selectiva doble tiene la condición $condición1$ y dentro de esta se tiene la segunda estructura algorítmica selectiva doble que posee la condición $condición2$.

En este caso si no se cumple la condición $condición1$ se ejecuta el conjunto de instrucciones c. Si se cumple la condición $condición1$ se evalúa la condición $condición2$ y dependiendo del valor de ésta se ejecuta el conjunto de instrucciones a o el conjunto de instrucciones b.

1.9.3. Representación en diagrama de flujo

En la figura 1.12 se puede apreciar la representación en diagram flujo correspondiente al pseudocódigo presentado en la figura 1.11.

1.9.4. Implementación en C

En el programa 1.9 se puede apreciar el mismo ejemplo presentado en las figuras 1.12 y 1.11 en C.

Si *condición1* **Entonces**
  Si *condición2* **Entonces**
    conjunto de instrucciones a;
  SiNo
    conjunto de instrucciones b;
  Fin Si
SiNo
  conjunto de instrucciones c;
Fin Si

Figura 1.11: Pseudocódigo: Estructuras selectivas anidadas

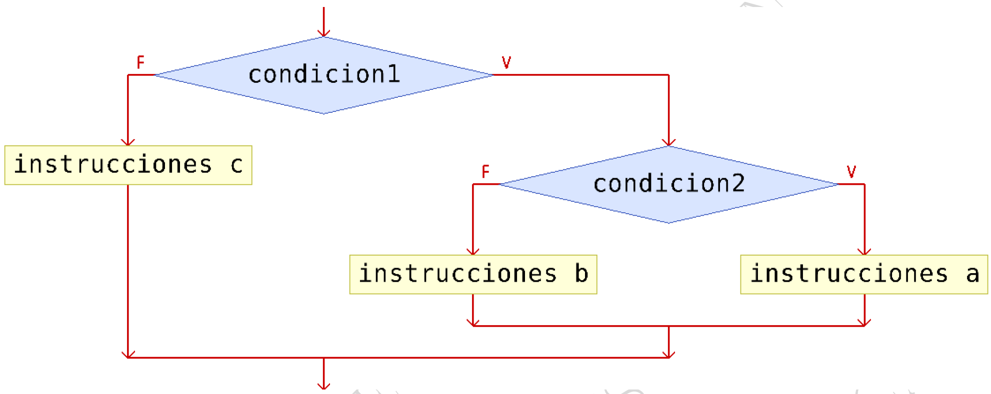

Figura 1.12: Diagrama de Flujo: Selectiva anidada

Recordar que:

El lenguaje C no implementa nativamente el tipo de dato lógico (`bool` o `boolean`). Entonces ,¿cómo hace el lenguaje C para evaluar la condición de la instrucción `if`?

- C asume que todo valor igual a 0 falla una condición. El comportamiento es muy similar al *falso* de una expresión lógica.
- Por el contrario, un valor diferente de 0 hará que se cumpla la condición. Comportamiento similar al *verdadero* de una expresión lógica.

Por ejemplo: si se tienen las siguientes definiciones `int suma=0, i=10;`, las siguientes expresiones serán consideradas *verdaderas*: $suma \leq 100$, $i == 10$, $suma < i$, i. Por otro lado, las siguientes expresiones serán consideradas *falsas*: $suma \geq 100$, $i == 20$, $suma > i$, $suma$.

Programa 1.9: C: Estructura selectiva anidada

```
...
if (condición1){
        if (condición2){
                conjunto de instrucciones a;
        }
        else{
                conjunto de instrucciones b;
        }
}
else{
            conjunto de instrucciones c;
}
```

```
...
```

1.10. Cálculo de las raíces ecuaciones cuadráticas con una variable - Selectivas Anidadas

Una ecuación cuadrática con una variable es una ecuación que tiene la forma de $ax^2 + bx + c = 0$ siendo a, b y c números reales con la restricción que $a \neq 0$. Para encontrar la solución a la ecuación se puede utilizar la siguiente fórmula general $x = \frac{-b\pm\sqrt{b^2-4ac}}{2a}$. La fórmula genera una solución con dos raíces, la $x_1 = \frac{-b+\sqrt{b^2-4ac}}{2a}$ y la $x_2 = \frac{-b-\sqrt{b^2-4ac}}{2a}$ las cuales no son necesariamente diferentes.

Dada una ecuación cuadrática con una variable se solicita que elabore un algoritmo expresado en diagrama de flujo y pseudocódigo así como un programa en C que calcule las raíces de la solución. En la guía #2 se resolvió este problema y se asumió que el discriminante siempre sería mayor o igual a 0 por lo que la solución daría un número real. En la guía #3 se verificó el discriminante para que calcule las raíces ya sean reales o complejas. ¿Cómo se puede adaptar el programa para que calcule las raíces reales y complejas distinguiendo además el caso cuando exista una única raíz real? Este es una situación ideal para una estructura selectiva doble anidada.

Se puede diseñar una estructura selectiva doble anidada para decidir sobre 3 caminos diferentes. El primero de ellos se va a dar cuando el discriminantes es igual a cero, en este caso existirá una única solución real. El segundo camino es cuando el discriminante es mayor que cero, en este caso existirán dos raíces reales. El tercer camino se dará cuando el discriminante sea menor que cero, en este caso existirán dos raíces complejas.

Recordar que:

En el contexto de los números complejos $i = \sqrt{-1}$

En la figura 1.13 se puede apreciar el diagrama de flujo en donde se diseña la alternativa de solución propuesta. Note la inclusión de la estructura selectiva anidada en la parte derecha del diagrama. El conjunto de instrucciones de la selectiva doble cuya condición es `discriminante` $\geq$ `0`, incluye otra estructura selectiva doble cuya condición es `discriminante = 0`.

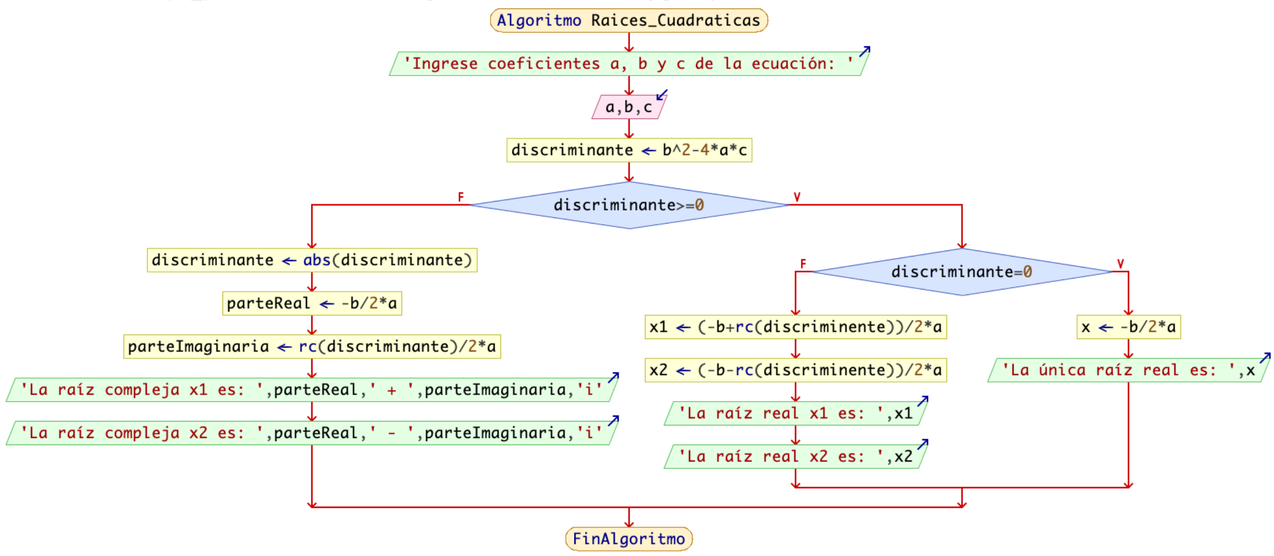

Figura 1.13: Diagrama de flujo: Ecuación cuadrática con raíces reales y complejas

En la figura 1.14 se aprecia el pseudocódigo de la misma alternativa de solución. Por razones estéticas, los conjuntos de instrucciones de la estructura selectiva doble se indentan. Note la inclusión de la estructura selectiva anidada entre las líneas 6 y 14. La estructura selectiva doble que incluye a la selectiva anidada la puede observar entre las líneas 5 y 21.d

Recordar que:

- La indentación es un sangría que se incluye en un conjunto de instrucciones. Todo el conjunto de instrucciones se mueve unos caracteres a la derecha. Si bien es cierto a las herramientas que ejecutan código no les afecta la indentación de código, a los humanos sí. Es mucho más fácil entender un código bien indentado que un código sin indentación. Esto afecta mucho la etapa de corrección de errores y al mantenimiento de software.
- Muchas veces el resultado de la comparación de números reales a través de la igualdad no es el deseado. En este caso es recomendable usar el valor absoluto de la diferencia de los números que se desean comparar. Si esta diferencia es cercana a cero (en esta ocasión será menor a 0.0001), se puede asumir que son iguales. Considere que en PSeInt se utiliza la función abs() para obtener el valor absoluto de un número.

```
1  Algoritmo Raices_Cuadraticas
2      Escribir 'Ingrese coeficientes a, b y c de la ecuación: '
3      Leer a,b,c
4      discriminante ← b↑2-4*a*c
5      Si discriminante ≥ 0 Entonces
6          Si abs(discriminante) ≤ 0.0001 Entonces
7              x ← -1*b/(2*a)
8              Escribir 'La única raíz real es: ', x
9          SiNo
10             x1 ← (-1*b + rc(discriminante))/(2*a)
11             x2 ← (-1*b - rc(discriminante))/(2*a)
12             Escribir 'La raíz real x1 es: ', x1
13             Escribir 'La raíz real x2 es: ', x2
14         FinSi
15     SiNo
16         discriminante ← abs(discriminante)
17         parteReal ← -1*b/(2*a)
18         parteImaginaria ← rc(discriminante)/(2*a)
19         Escribir 'La raíz compleja x1 es: ', parteReal, ' + ', parteImaginaria, 'i'
20         Escribir 'La raíz compleja x2 es: ', parteReal, ' - ', parteImaginaria, 'i'
21     FinSi
22 FinAlgoritmo
```

Figura 1.14: Pseudocódigo: Ecuación cuadrática con raíces reales y complejas

En el programa 1.3 se puede apreciar la alternativa de solución en C. Al igual que en el pseudocódigo, el conjunto de instrucciones de las selectivas se encuentran indentados.

Programa 1.10: Ecuación cuadrática con raíces reales y complejas

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

int main() {
    double a, b, c;
    double discriminante;

    printf("Ingrese coeficientes a, b y c de la ecuación: ");
    scanf(" %lf %lf %lf", &a, &b, &c);

    discriminante = pow(b, 2) - 4 * a*c;
    if (discriminante >= 0) {
        if (fabs(discriminante) <= 0.0001) {
            double x;
            x = (-1*b + sqrt(discriminante)) / (2 * a);
            printf("La única raíz real es %lf\n", x);
        } else {
            double x1, x2;
            x1 = (-1*b + sqrt(discriminante)) / (2 * a);
            x2 = (-1*b - sqrt(discriminante)) / (2 * a);
            printf("La raíz real x1 es %lf\n", x1);
            printf("La raíz real x2 es %lf\n", x2);
        }
    } else {
        double parteReal, parteImaginaria;
        discriminante = fabs(discriminante);
        parteReal = -1*b / 2 * a;
        parteImaginaria = sqrt(discriminante) / (2 * a);
        printf("La raíz compleja x1 es %.2lf + %.2lfi\n", parteReal, parteImaginaria);
        printf("La raíz compleja x2 es %.2lf - %.2lfi\n", parteReal, parteImaginaria);
    }
    return 0;
}
```

A continuación sigue un ejemplo de ejecución de este programa

```
Ingrese coeficientes a, b y c de la ecuación: 1 -4 13
La raíz compleja x1 es 2.00 + 3.00i
La raíz compleja x2 es 2.00 - 3.00i
```

Para poner en práctica

¿Cuáles son las raíces de las siguientes ecuaciones?

- $x^2 + x + 1 = 0$
- $-2x^2 + 3x + 4 = 0$
- $x^2 + 3x + 6 = 0$
- $-x^2 - 2x + 4 = 0$
- $2x^2 + 3x + 4 = 0$

1.11. ¿Cuántos dígitos tiene un número?

¿Cómo saber cuántos dígitos tiene un número? Podemos usar las matemáticas para resolver este problema. Se sabe que para determinado número n se cumple que $10^{k-1} \leq n < 10^k$, donde k representa el número de cifras de dicho número. Por ejemplo si $n = 253$ entonces $10^2 < 253 < 10^3$, en este caso 3 es el número de cifras.

Para obtener el valor de k se aplica logaritmo en base 10 y se obtiene que $k-1 \leq log_{10}(n) < k$. De esta última relación se deduce que $k-1 \leq log_{10}(n)$ por lo que $k \leq 1+log_{10}(n)$, como k debe ser un número entero entonces

$k = 1 + \lfloor log_{10}(n) \rfloor$. Utilizaremos esta relación matemática para calcular el número de dígitos dado un número n dado como parámetro.

Recuerde que para usar los logaritmos el argumento debe ser mayor que 0. Entonces, ¿cómo hacemos para calcular el número de dígitos de un número negativo?, ¿cómo hacemos para calcular el número de dígitos cuando ingresen 0? Como se tienen varias condiciones y diferentes respuestas para cada interrogante, procederemos a utilizar una estructura algorítmica selectiva doble anidada. ¿Cuál sería la mejor condición para la selectiva doble? Luego de analizar el problema, se ve que se puede colocar como condición `numero=0`, si se cumple esta condición es fácil calcular el número de dígitos, será siempre 1. Si no se cumple la condición podemos utilizar la relación matemática analizada previamente. Pero hay un detalle, que la relación matemática solo funciona para números positivos, entonces, si es que el número es negativo utilizaremos su valor absoluto. Esto lo realizaremos con la estructura algorítmica selectiva anidada.

En la figura 1.15 se puede apreciar una propuesta de solución al problema propuesto expresado en diagrama de flujo. En la figura 1.16 se aprecia la misma propuesta de solución expresada en pseudocódigo. En ambas soluciones se ha utilizado la función `trunc` que toma la parte entera de un número real. En PSeInt no existe la función de logaritmo en base 10, para hacer este cálculo se tiene que aplicar el teorema de cambio de base de los logaritmos. PSeInt implementa el logaritmo natural.

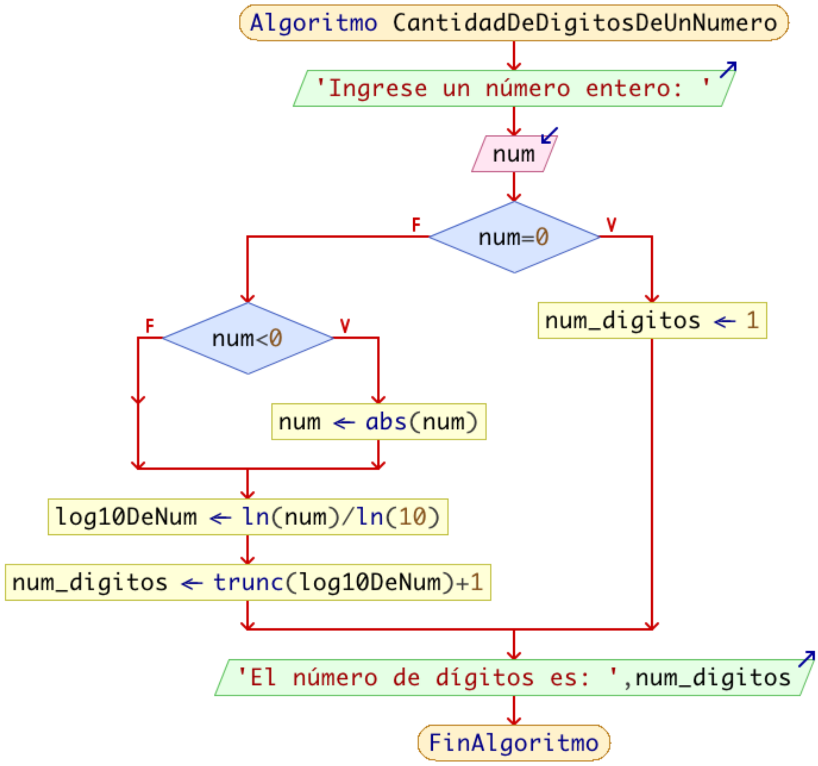

Figura 1.15: Diagrama de flujo: Conteo de dígitos de un número

En el programa 1.11 se puede apreciar la alternativa de solución en un programa en C. Hay tres aspectos vale la pena analizar detenidamente. En primer lugar, la función para calcular el valor absoluto de un número entero es `abs` cuya declaración se encuentra en el archivo de cabecera `stdlib`. Note la diferencia con el programa 1.3 en donde se utilizó la función `fabs` cuya declaración se encuentra en el archivo de cabecera `math.h`. Se utilizó esta función pues el argumento era un número real.

El segundo aspecto a considerar es el uso de la función `log10` cuya declaración se encuentra en el archivo de cabecera `math.h`. A diferencia de PSeInt, C sí define este logaritmo. En C el logaritmo natural se implementa a través de la función `log`.

El tercer aspecto a considerar es el truncamiento. Existe en C la función `trunc` pero esta no ha sido empleada

```
Algoritmo CantidadDeDigitosDeUnNumero
    Escribir 'Ingrese un número entero: '
    Leer num
    Si num=0 Entonces
        num_digitos <- 1
    SiNo
        Si num<0 Entonces
            num <- abs(num)
        FinSi
        log10DeNum <- ln(num)/ln(10)
        num_digitos <- trunc(log10DeNum)+1
    FinSi
    Escribir 'El número de dígitos es: ',num_digitos
FinAlgoritmo
```

Figura 1.16: Pseudocódigo: Conteo de dígitos de un número

en el programa por que al ser la variable `num_digitos` entera, al asignársele un valor real, el C automáticamente hace un truncamiento.

Programa 1.11: Conteo de dígitos de un número

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

int main() {
    int numero, num_digitos;

    printf("Ingrese un número entero: ");
    scanf(" %d", &numero);

    if (numero==0)
        num_digitos = 1;
    else{
        if (numero<0)
            numero = abs(numero);
        num_digitos = log10(numero) + 1;
    }
    printf("El número de dígitos es: %d\n", num_digitos);
    return 0;
}
```

1.12. Impresión en diferentes bases

Se desea implementar un programa en C que dado un número n entero expresado en base 10 y una base, se imprima el número n pero expresado en la base ingresada por el usuario. La base podrá ser 8 o 16. En caso el usuario ingrese otra base, el programa deberá emitir un mensaje de error.

Para solucionar este problema utilizaremos una de las características de formato de la función `printf`. Si en el formato de la función `printf` se coloca `%o` se imprime el número en octal, si se coloca `%x` se imprime el número en hexadecimal[4]. Una alternativa de solución a este problema se puede apreciar en el programa 1.12. En este programa, en el conjunto de instrucciones que corresponde cuando no se cumple la condición (líneas 13 – 17), se encuentra en una selectiva doble anidada.

Programa 1.12: Impresión en diferentes bases

```c
1 #include <stdio.h>
2
```

[4]Si se utiliza `%X` los dígitos hexadecimales se imprimirán en mayúscula.

```c
int main() {
    int numero, base;

    printf("Ingrese número en base 10: ");
    scanf(" %d", &numero);
    printf("Ingrese base a transformar (8 o 16): ");
    scanf(" %d", &base);

    if (base == 8)
        printf("El número en base 8 es %o\n", numero);
    else
        if (base == 16)
            printf("El número en base 16 es %x\n", numero);
        else
            printf("Base no soportada.\n");
    return 0;
}
```

1.13. Funciones definida por tramos

Otra situación en donde se pueden aplicar estructuras selectivas dobles anidadas es en las funciones definidas por tramos. A continuación se presenta una función de este tipo:

$$f(x) = \begin{cases} x-3 & x \leq 1 \\ x^2+1 & 1 < x \leq 3 \\ 11 & x > 3 \end{cases}$$

Como se puede apreciar en la función $f(x)$, dependiendo del rango de valores, se realiza determinada expresión matemática. Es decir dependiendo de la condición (rango de valores), se decide un flujo de ejecución a realizar (expresión matemática). Una posible solución a este problema es utilizar como primera condición en una selectiva doble, la verificación del primer rango, esto se puede realizar usando la condición `x <= 1`, si la condición se cumple, se realiza el cálculo correspondiente. Si no se cumple, solo quedan dos condiciones a probar. Para estas dos últimas condiciones, se utiliza otra selectiva doble anidada.

En la figura 1.17 se puede apreciar una alternativa de solución a la función definida por tramos dado determinado número x. El algoritmo se encuentra expresado en diagrama de flujo. En la figura 1.18 se aprecia la misma solución en pseudocódigo. La implementación de esta alternativa de solución en C se puede apreciar en el programa 1.13.

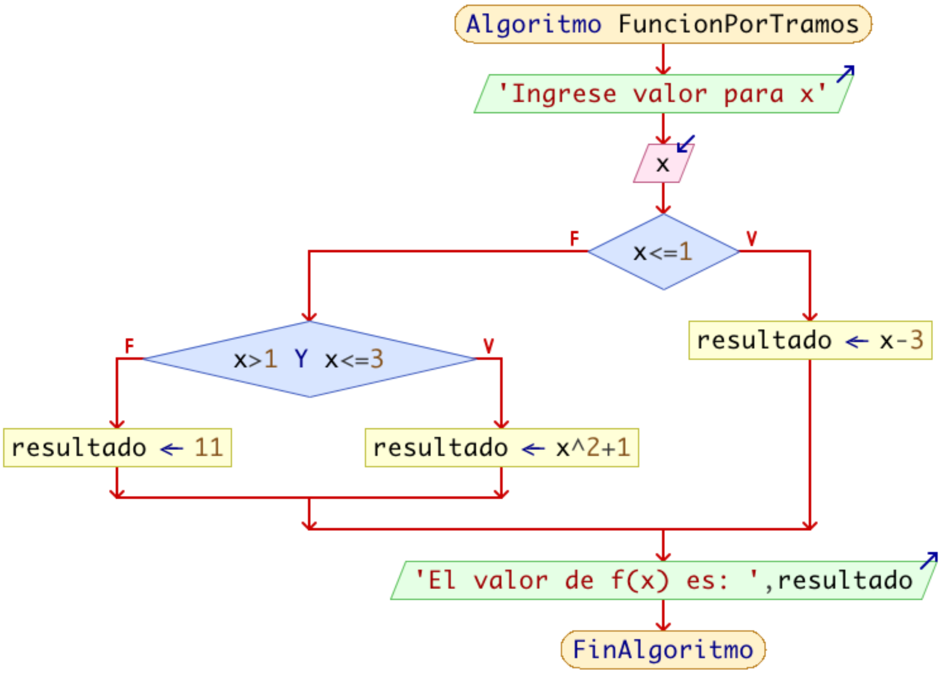

Figura 1.17: Diagrama de flujo: Función definida por tramos

```
Algoritmo FuncionPorTramos
    Escribir 'Ingrese valor para x'
    Leer x
    Si x<=1 Entonces
        resultado <- x-3
    SiNo
        Si x>1 y x<=3 Entonces
            resultado <- x^2+1
        SiNo
            resultado <- 11
        FinSi
    FinSi
    Escribir 'El valor de f(x) es: ', resultado
FinAlgoritmo
```

Figura 1.18: Pseudocódigo: Función definida por tramos

Programa 1.13: Función definida por tramos

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

int main() {
    double x, y;

    printf("Ingrese el valor para x: ");
    scanf(" %lf", &x);

    if (x <= 1)
        y = x-3;
    else
        if (x>1 && x<=3)
            y=pow(x,2)+1;
        else
            y=11;
    printf("El valor de f(x) es %lf\n", y);
    return 0;
}
```

Capítulo 2

Ejercicios propuestos

2.1. Nivel Básico

En la pregunta 2.1.1 se presenta una serie de programas sintácticamente correctos pero que contienen errores lógicos comunes en el uso de operadores y la estructura algorítmica selectiva en el lenguaje C. Para cada uno de estos programas se le solicita que los analice y reflexione sobre el resultado que debería obtener. Luego ejecute el programa usando un Entorno de Desarrollo Integrado (IDE) de lenguaje C para verificar el resultado.

Para los demás ejercicios propuestos, se solicita que elabore el correspondiente algoritmo representado tanto en diagrama de flujo como en pseudocódigo así como la implementación de un programa en lenguaje C conforme a los temas revisados en las guías *preliminar* y #1 del curso Fundamentos de Programación.

2.1.1. Análisis de programas

Programa 2.1: Análisis de programa: Operador de asignación

```c
1 #include <stdio.h>
2
3 int main() {
4     int a=10;
5     if (a=5)
6         printf("El valor de a= %d\n", a);
7     return 0;
8 }
```

Para analizar

- ¿El valor que se imprime es el valor esperado?
- ¿Qué hace la línea 5?
- Cambie la línea 5 por `if(a==5)`. ¿Qué cambia en el programa?

Programa 2.2: Análisis de programa: Operador relacional de comparación

```c
#include <stdio.h>

int main() {
    int a=10, b=10, c=10;
    if (a==b==c)
        printf("Los 3 números son iguales\n");
    return 0;
}
```

Para analizar

- ¿El valor que se imprime es el valor esperado?
- ¿El lenguaje C permite la comparación múltiple?
- Si quisiéramos verificar que los 3 números sean iguales, ¿cómo debería ser la condición de la estructura selectiva `if` en la línea 5?

Programa 2.3: Análisis de programa: Operador relacional de comparación

```c
#include <stdio.h>

int main() {
    int a=1, b=1, c=1;
    if (a==b==c)
        printf("Los 3 números son iguales\n");
    return 0;
}
```

Para analizar

- ¿Por qué en este programa pareciera que el lenguaje C sí implementa la comparación múltiple?

Programa 2.4: Análisis de programa: Operador aritmético de suma

```c
#include <stdio.h>

int main() {
    int a=5, b=-5;
    if (a+b)
        printf("La suma es cero\n");
    return 0;
}
```

Para analizar

- ¿Cuál es el error lógico en este programa?
- Corrija el error lógico usando el operador de comparación (`==`).
- Corrija el error lógico usando el operador de negación (`!`).

Programa 2.5: Análisis de programa: Operador aritmético de suma

```c
1 #include <stdio.h>
2 
3 int main() {
4     int a=-3, b=-5;
5     if (a+b)
6         printf("La suma es diferente de cero\n");
7     return 0;
8 }
```

Para analizar

- ¿Este programa tiene algún error lógico?
- ¿Es necesario cambiar la condición de la línea 5 por `(a+b)!=0`?
- Si a la variable `b` se le hubiera asignado el valor de 3. ¿Qué se imprime?

2.1.2. El valor absoluto

Dado determinado número real, se le pide que retorne el valor absoluto de dicho número.

Recordar que:

El valor absoluto de un número real x, es la magnitud numérica de dicho número sin importar su signo.

$$\mid x \mid = \begin{cases} x & \text{si } x \geq 0 \\ -x & \text{si } x < 0 \end{cases}$$

Restricciones

- Para la implementación en lenguaje C no podrá usar la función `fabs`.
- Para la implementación en PSeInt no podrá usar la función `abs`.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $x = -1.6$, entonces se debe imprimir 1.6.
- Si $x = 2.5$, entonces se debe imprimir 2.5.
- Si $x = -3.9$, entonces se debe imprimir 3.9.
- Si $x = 6.3$, entonces se debe imprimir 6.3.

2.1.3. La función techo

Dado determinado número real, se le pide que retorne el techo de dicho número.

Recordar que:

La función techo es una función que recibe como parámetro a un número real x ($x \in \mathbb{R}$) y retorna el mínimo entero y ($y \in \mathbb{Z}$) más próximo que es mayor o igual a x.

$$techo(x) = \lceil x \rceil = min\{k \in \mathbb{Z} | x \leq k\}$$

Sugerencia

Realice el siguiente procedimiento:

- Lea el número real x en una variable.
- Obtenga la parte entera del número real x usando la función `trunc`. Almacene este valor en una variable. En lenguaje C la declaración de la función `trunc` se encuentra en el archivo de cabecera `math.h`.
- Obtenga la parte fraccionaria del número real x. Para esto reste a x la parte entera hallada en el paso anterior. Almacene este valor en una variable.
- Si la parte fraccionaria del número real x es mayor que cero, incremente en uno el valor de la parte entera.
- El techo de x se encontrará en la variable en donde se almacenó la parte entera.

Restricciones

- Para la implementación en lenguaje C no podrá usar la función `ceil`.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $x = -3.4$, entonces se debe imprimir -3.
- Si $x = -0.7$, entonces se debe imprimir 0.
- Si $x = 0$, entonces se debe imprimir 0.
- Si $x = 1.4$, entonces se debe imprimir 2.
- Si $x = 3.8$, entonces se debe imprimir 4.
- Si $x = 4.1$, entonces se debe imprimir 5.

2.1.4. La función piso

Dado determinado número real, se le pide que retorne el piso de dicho número.

Recordar que:

La función piso es una función que recibe como parámetro a un número real x ($x \in \mathbb{R}$) y retorna el máximo entero y ($y \in \mathbb{Z}$) más próximo que es menor o igual a x.

$$piso(x) = \lfloor x \rfloor = max\{k \in \mathbb{Z} | k \leq x\}$$

Sugerencia

Realice el siguiente procedimiento:

- Lea el número real x en una variable.
- Si el número real x es mayor o igual a cero, la función piso se obtiene aplicando la función `trunc`. En lenguaje C la declaración de la función `trunc` se encuentra en el archivo de cabecera `math.h`.
- Si el número real x es menor que cero:
  - Obtenga la parte entera del número real x usando la función `trunc`. Almacene este valor en una variable.
  - Obtenga la parte fraccionaria del número real x. Para esto reste a x la parte entera hallada en el paso anterior. Almacene este valor en una variable.
  - Si la parte fraccionaria del número real x es diferente de cero, decremente en uno el valor de la parte entera.
  - El piso de x se encontrará en la variable en donde se almacenó la parte entera.

Restricciones

- Para la implementación en lenguaje C no podrá usar la función `floor`.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $x = -3.4$, entonces se debe imprimir -4.
- Si $x = -0.7$, entonces se debe imprimir -1.
- Si $x = 0$, entonces se debe imprimir 0.
- Si $x = 1.4$, entonces se debe imprimir 1.
- Si $x = 3.8$, entonces se debe imprimir 3.
- Si $x = 4.1$, entonces se debe imprimir 4.

2.1.5. ¿Cuántos días tiene determinado año?

Dado determinado número entero que representa un año, imprima la cantidad de días que posee dicho año. Recuerde que los años bisiestos tienen un día adicional, el 29 de febrero.

Recordar que:

Un año es bisiesto si el número que lo representa es divisible entre 4, salvo que sea año secular –último de cada siglo, terminado en 00–, en cuyo caso también ha de ser divisible entre 400.

Dadas las siguientes preposiciones:

- p: El número que representa al año es divisible entre 4.
- q: El número que representa al año es divisible entre 100.
- r: El número que representa al año es divisible entre 400.

La expresión lógica que permite determinar si un año es bisiesto es $p \wedge (\neg q \vee r)$.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $año = 2013$, deberá imprimir 365.
- Si $año = 2016$, deberá imprimir 366.
- Si $año = 2019$, deberá imprimir 365.
- Si $año = 2020$, deberá imprimir 366.
- Si $año = 2024$, deberá imprimir 366.
- Si $año = 2028$, deberá imprimir 366.

2.1.6. ¿Cuántos paquetes utilizar?

Se requiere colocar cierta cantidad l de lapiceros en paquetes de capacidad c. ¿Cuántos paquetes se deberán utilizar para este fin?

Restricciones

- Para la implementación en lenguaje C no podrá usar la función `ceil`.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $l = 12$ y $c = 15$, deberá retornar `Se requiere 1 paquete(s)`.
- Si $l = 20$ y $c = 6$, deberá retornar `Se requiere 4 paquete(s)`.
- Si $l = 25$ y $c = 5$, deberá retornar `Se requiere 5 paquete(s)`.
- Si $l = 30$ y $c = 4$, deberá retornar `Se requiere 8 paquete(s)`.

2.1.7. ¿Cuánto cuesta la matrícula?

En determinada universidad la matrícula del semestre tiene un costo de c soles. A los alumnos que se encuentran en facultad se les realiza un descuento de $d\,\%$. Se le solicita que dado el costo c de la matrícula, el descuento d y el número n del ciclo en que se encuentra matriculado determinado alumno, calcule y presente el monto que debe pagar dicho alumno. Asuma que los alumnos se encuentran en facultad si es que el ciclo en que se encuentra es mayor o igual a 5.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $c = 1500$, $d = 15\,\%$ y $n = 2$, deberá imprimir `Deberá pagar 1500 soles`.
- Si $c = 1500$, $d = 15\,\%$ y $n = 5$, deberá imprimir `Deberá pagar 1275 soles`.
- Si $c = 2000$, $d = 25\,\%$ y $n = 4$, deberá imprimir `Deberá pagar 2000 soles`.
- Si $c = 2000$, $d = 25\,\%$ y $n = 8$, deberá imprimir `Deberá pagar 1500 soles`.

2.1.8. Longitud de onda

La nota musical `LA` tiene una frecuencia, por convenio internacional, de 440 Hz. Esta nota se propaga con una velocidad de 340 m/s en el aire y con una velocidad de 1400 m/s en el agua. Se le pide que lea un caracter que represente al medio en donde se propaga la nota musical (`A` para el aire y `G` para el agua) e imprima la longitud de la onda en dicho medio.

Recordar que:

$$\lambda = \frac{v}{f}$$

donde:

- λ es la longitud de onda.
- v es la velocidad de propagación.
- f es la frecuencia de la onda.

Además: $1\ Hz = \frac{1}{s}$

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si el medio ingresado es `A`, entonces deberá retornar $\lambda_{agua} \approx 3.181818182$ m
- Si el medio ingresado es `G`, entonces deberá retornar $\lambda_{aire} \approx 0.772727273$ m

2.1.9. El mayor de 3 números

Dado 3 números diferentes entre sí, se les pide que encuentre el mayor de ellos.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $a = -1$, $b = 4$ y $c = 3$ entonces se debe imprimir `El mayor número es 4`.
- Si $a = -1$, $b = 4$ y $c = 4$ entonces se debe imprimir `Los 3 números deben ser diferentes`.
- Si $a = 10$, $b = -6$ y $c = -9$ entonces se debe imprimir `El mayor número es 10`.

Sugerencia

Realice las siguientes modificaciones al problema:

- En lugar de encontrar el mayor de 3 números encuentre el menor.
- Determine si los números ingresados están en orden ascendente. Por ejemplo si $a = -1$, $b = 4$ y $c = 5$, deberá imprimirse `Los números están en orden ascendente`.
- Determine si los números ingresados están en orden descendente. Por ejemplo si $a = 10$, $b = -6$ y $c = -9$, deberá imprimirse `Los números están en orden descendente`.

2.1.10. Cantidad de días por mes

Dado un determinado número de mes y un determinado año, se solicita que calcule la cantidad de días que existen en dicho mes.

Recordar que:

Los meses poseen la siguientes cantidades de días.

- Enero, Marzo, Mayo, Julio, Agosto, Octubre, Diciembre poseen 31 días.
- Abril, Junio, Septiembre, Noviembre poseen 30 días.
- Febrero posee 28 días. Salvo los años bisiestos en donde posee 29 días.

Recordar que:

Un año es bisiesto si el número que lo representa es divisible entre 4, salvo que sea año secular –último de cada siglo, terminado en 00–, en cuyo caso también ha de ser divisible entre 400.

Dadas las siguientes preposiciones:

- p: El número que representa al año es divisible entre 4.
- q: El número que representa al año es divisible entre 100.
- r: El número que representa al año es divisible entre 400.

La expresión lógica que permite determinar si un año es es $p \wedge (\neg q \vee r)$.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $mes = 11$ y $año = 2013$, entonces se debe imprimir 30.
- Si $mes = 2$ y $año = 2016$, entonces se debe imprimir 29.
- Si $mes = 1$ y $año = 2017$, entonces se debe imprimir 31.
- Si $mes = 2$ y $año = 2019$, entonces se debe imprimir 28.
- Si $mes = 7$ y $año = 2020$, entonces se debe imprimir 31.
- Si $mes = 6$ y $año = 2024$, entonces se debe imprimir 30.

2.1.11. Calculadora de operaciones lógicas

Dado dos caracteres que representan el valor de una proposición lógica (`V` o `F`) y un caracter que representa una operación lógica (C que representa a la operación de conjunción, D que representa a la operación de disyunción, K que representa a la operación de condicional y B que representa a la operación biocondicional), se solicita que retorne el resultado de aplicar la operación lógica a los valores dados. En caso que los valores dados y la operación ingresada no corresponda con las antes mencionadas, deberá emitirse un mensaje de advertencia y no realizar las operaciones.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $p = V$, $q = V$ y $operación = C$ se debe imprimir `El resultado es V`.
- Si $p = V$, $q = F$ y $operación = C$ se debe imprimir `El resultado es F`.
- Si $p = F$, $q = F$ y $operación = D$ se debe imprimir `El resultado es F`.
- Si $p = V$, $q = F$ y $operación = D$ se debe imprimir `El resultado es V`.
- Si $p = V$, $q = F$ y $operación = K$ se debe imprimir `El resultado es F`.
- Si $p = F$, $q = F$ y $operación = B$ se debe imprimir `El resultado es V`.

2.1.12. Conversión de grados

Dado la magnitud de un grado y un caracter que representa la unidad de medida del grado (`R` para radianes y `S` para sexagesimal), se requiere que imprima la magnitud del grado pero expresado en la otra unidad de medida del grado, es decir si el usuario ingresa radianes, se deberá imprimir el grado en sexagesimales, pero si el usuario ingresa grados sexagesimales, se deberá imprimir el mismo en radianes.

Recordar que:

$$360^\circ = 2\pi \text{ radianes}$$

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $magnitud = 45$ y $unidad = S$ se debe imprimir `45 grados sexagesimales equivale a 0.78 radianes`.
- Si $magnitud = 55$ y $unidad = S$ se debe imprimir `55 grados sexagesimales equivale a 0.95 radianes`.
- Si $magnitud = 1.570796327$ y $unidad = R$ se debe imprimir `1.570796327 radianes equivale a 90 grados sexagesimales`.
- Si $magnitud = 0.925024504$ y $unidad = R$ se debe imprimir `0.925024504 radianes equivale a 53 grados sexagesimales`.

2.1.13. Conversión de temperatura

Dado la magnitud de una temperatura, un caracter que representa la unidad de medida de la magnitud de la temperatura (`F` para Fahrenheit, `C` para centígrados y `K` para Kelvin) y un caracter que representa la unidad de medida de la temperatura a la cual se desea convertir (`F` para Fahrenheit, `C` para centígrados y `K` para Kelvin), se requiere que imprima la magnitud de la temperatura expresada en la unidad de medida deseada.

Recordar que:

- $F = \frac{5}{9}C + 32$
- $K = C + 273.15$

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $magnitud = 23$, $unidad = C$ y $unidad\ de\ conversion = C$ se debe imprimir `No es necesario hacer la conversión`.
- Si $magnitud = 23$, $unidad = C$ y $unidad\ de\ conversion = F$ se debe imprimir `Equivale a 73.4`.
- Si $magnitud = 23$, $unidad = C$ y $unidad\ de\ conversion = K$ se debe imprimir `Equivale a 296.15`.
- Si $magnitud = 270$, $unidad = K$ y $unidad\ de\ conversion = F$ se debe imprimir `Equivale a 26.33`.
- Si $magnitud = 80$, $unidad = F$ y $unidad\ de\ conversion = K$ se debe imprimir `Equivale a 299.81`.

2.1.14. Aplicación de la ley de Boyle

La ley de Boyle establece una relación entre la presión y el volumen de un gas, de forma tal que si la presión cambia, se puede calcular el volumen resultante. De forma análoga si el volumen cambia, es posible también calcular la temperatura.

Recordar que:

La ley de Boyle establece que la presión de un gas en un recipiente cerrado es inversamente proporcional al volumen del recipiente, cuando la temperatura permanece constante.

De esta ley se obtiene la siguiente relación: $P_1 \times V_1 = P_2 \times V_2$

Se le pide que dado el valor que se desea hallar (e.g. P_1, P_2, V_1, V_2), solicite al usuario los valores restantes y calcule el valor deseado. Por ejemplo si el usuario desea hallar el valor de P_1, deberá solicitar los valores de P_2, V_1 y V_2. Si el usuario desea hallar el valor de V_1, deberá solicitar los valores de P_1, P_2 y V_2. Asuma que los todos los valores de presión se expresan en atm y el volumen en cm^3.

Casos de prueba

Use los siguientes datos para probar su solución:

- Si el dato a hallar es V_1 y el usuario ingresa $P_1 = 0.98\ atm$, $P_2 = 1.2\ atm$ y $V_2 = 65.78\ cm^3$, entonces deberá imprimir `El valor de V1 es 80`.
- Si el dato a hallar es P_1 y el usuario ingresa $V_1 = 90\ cm^3$, $P_2 = 1.5\ atm$ y $V_2 = 55.26\ cm^3$, entonces deberá imprimir `El valor de P1 es 0.92`.
- Si el dato a hallar es P_2 y el usuario ingresa $V_1 = 100\ cm^3$, $P_1 = 0.86\ atm$ y $V_2 = 77.75\ cm^3$, entonces deberá imprimir `El valor de P2 es 1.1`.

2.1.15. Aplicación de la ley de Charles

La ley de Charles establece una relación entre el volumen y la temperatura de un gas, de forma tal que si el volumen cambia, se puede calcular la temperatura resultante. De forma análoga si la temperatura cambia, es posible también calcular el volumen.

Recordar que:

Ley de Charles establece que el volumen es directamente proporcional a la temperatura del gas, cuando la presión permanece constante. Si la temperatura aumenta, el volumen también aumenta. Si la temperatura disminuye, el volumen también disminuye.

De esta ley se obtiene la siguiente relación: $\frac{V_1}{T_1} = \frac{V_2}{T_2}$

Se le pide que dado el valor que se desea hallar (e.g. V_1, V_2, T_1, T_2), solicite al usuario los valores restantes y calcule el valor deseado. Por ejemplo si el usuario desea hallar el valor de V_1, deberá solicitar los valores de V_2, T_1 y T_2. Si el usuario desea hallar el valor de T_1, deberá solicitar los valores de V_1, V_2 y T_2. Asuma que los todos los valores de volumen se expresan en cm^3 y la temperatura en °.

Casos de prueba

Use los siguientes datos para probar su solución:

- Si el dato a hallar es V_1 y el usuario ingresa $T_1 = 20$ °C y $T_2 = 90$ °C y $V_2 = 900\ cm^3$, entonces deberá imprimir `El valor de V1 es 200`.
- Si el dato a hallar es T_1 y el usuario ingresa $V_1 = 100\ cm^3$, $T_2 = 75$ °C y el $V_2 = 214.28\ cm^3$, entonces deberá imprimir `El valor de T1 es 35`.
- Si el dato a hallar es $V2$ y el usuario ingresa $V_1 = 300\ cm^3$, $T_1 = 80$ °C y $T_2 = 35$ °C, entonces deberá imprimir `El valor de V2 es 131.25`.

2.1.16. La escalas de Wechsler

La Escala Wechsler de Inteligencia para Adultos (WAIS) fue desarrollada por primera vez en 1939 y fue llamada entonces el *Wechsler-Bellevue Intelligence Test*[1]. En esta escala se definen categorías nominales que, dependiendo del valor de coeficiente intelectual (CI), permiten definir la inteligencia. En la tabla 2.1 se pueden apreciar las escalas nominales de esta prueba de inteligencia.

Tabla 2.1: La escalas de Wechsler

Rango CI	Descripción
≥ 130	Muy superior
$[120 - 129]$	Superior
$[110 - 119]$	Sobre el promedio
$[90 - 109]$	Promedio
$[80 - 89]$	Bajo el promedio
$[70 - 79]$	Limítrofe
≤ 69	Muy bajo

Se pide que dado un valor de coeficiente intelectual, retorne la escala nominal que le corresponde en la Escala Wechsler.

2.1.17. Cálculo de la jornada de trabajo semanal

Se desea calcular el pago de la jornada de trabajo de determinado trabajador. A los trabajadores se les paga determinado monto por hora laborada, pero cuando la cantidad de horas excede a las 40 horas semanales, se le debe de pagar el 80 % por hora adicional. La jornada de trabajo regular es de lunes a viernes por lo que si se realizan horas adicionales los fines de semana, se le debe pagar el doble al trabajador por dichas horas.

[1] https://comenio.files.wordpress.com/2007/09/weshler_imprimir.pdf

Se le pide que dado el pago por hora de determinado trabajador, la cantidad de horas trabajadas de lunes a viernes y la cantidad de horas trabajadas el fin de semana, determine el pago que le corresponde a dicho trabajador.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si monto por hora es 13, cantidad de horas trabajadas de lunes a viernes es 30, cantidad de horas trabajadas el fin de semana es 0 entonces se debe imprimir `Se debe pagar 390 soles`.
- Si monto por hora es 15, cantidad de horas trabajadas de lunes a viernes es 40, cantidad de horas trabajadas el fin de semana es 0 entonces se debe imprimir `Se debe pagar 600 soles`.
- Si monto por hora es 12, cantidad de horas trabajadas de lunes a viernes es 55, cantidad de horas trabajadas el fin de semana es 10 entonces se debe imprimir `Se debe pagar 1044 soles`.
- Si monto por hora es 14, cantidad de horas trabajadas de lunes a viernes es 45, cantidad de horas trabajadas el fin de semana es 15 entonces se debe imprimir `Se debe pagar 1106 soles`.

2.2. Nivel Intermedio

2.2.1. Identidades trigonométricas

Se desea probar algunas identidades trigonométricas a través del computador. Para esto deberá de leer un ángulo en sexagesimal y luego calcular el valor de la identidad con el ángulo ingresado. El programa deberá imprimir `Se cumple identidad` en caso se cumpla la identidad trigonométrica y `No se cumple identidad` en caso contrario. Las identidades a probar serán las siguientes:

- $sen(2\theta) = 2sen(\theta)cos(\theta)$
- $cos(2\theta) = cos^2(\theta) - sen^2(\theta)$
- $cos(2\theta) = 1 - 2sen^2(\theta)$

Recordar que:

$360° = 2\pi$ radianes

Sugerencia para PSeInt

- Utilice la función `sen` para obtener el valor del seno un ángulo en radianes.
- Utilice la función `cos` para obtener el valor del coseno un ángulo en radianes.

Sugerencia para lenguaje C

- Utilice la función `sin` cuyo prototipo se encuentra en el archivo de cabecera `math.h` para obtener el valor del seno un ángulo en radianes.
- Utilice la función `cos` cuyo prototipo se encuentra en el archivo de cabecera `math.h` para obtener el valor del coseno un ángulo en radianes.

Comparación de números reales

Muchas veces el resultado de la comparación de números reales a través de la igualdad no es el deseado. Esto sucede por la forma en que se representa internamente los reales, basta que exista una pequeña diferencia de precisión para que no se de la igualdad. En este caso es recomendable usar el valor absoluto de la diferencia de los números que se desean comparar. Si esta diferencia es cercana a cero, se puede asumir que son iguales.

Casos de prueba

Deberá retornar una salida similar a la siguiente:

```
Se cumple la identidad sen(2x)=2sen(x)cos(x)
Se cumple la identidad cos(2x)=cos^2(x)-sen^2(x)
Se cumple la identidad cos(2x)=1-2sen^2(x)
```

2.2.2. La serie de Gregory

Una de las formas de calcular el número π es mediante la serie de Gregory, según:

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \cdots = \sum_{n=1}^{\infty} (-1)^{(n+1)} \frac{1}{2n-1}$$

Se le pide que dado un número n que representa el número del término de la serie, retorne dicho término.

Restricciones

- Para la implementación en lenguaje C no podrá usar la función `pow`.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $n = 1$, se deberá retornar 1
- Si $n = 3$, se deberá retornar 0.2
- Si $n = 4$, se deberá retornar ≈ -0.142857143
- Si $n = 9$, se deberá retornar ≈ 0.058823529

2.2.3. Suma de los cuadrados de los números en un rango

Se desea calcular la suma de los cuadrados de los números naturales que existen en un rango $[a..b]$ en donde tanto a como b son números naturales mayores que 0 y se debe cumplir además que $a < b$. Por ejemplo si $a = 5$ y $b = 10$, se deberá retornar $sumatoria = 5^2 + 6^2 + 7^2 + 8^2 + 9^2 + 10^2 = 25 + 36 + 49 + 64 + 81 + 100 = 355$.

Recordar que:

La suma de los cuadrados de los n primeros números naturales se puede calcular mediante la siguiente serie notable:

$$\sum_{i=1}^{n} i^2 = 1^2 + 2^2 + 3^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$$

Sugerencia

- Utilice una estructura algorítmica selectiva para verificar que los valores de a y b cumplen la condición del problema.
- Utilice la serie notable para calcular la suma de los cuadrados de los n primeros números naturales tomando como n el valor de b.
- Utilice la serie notable para calcular la suma de los cuadrados de los n primeros números naturales tomando como n el valor de $a - 1$.
- Obtenga la diferencia entre ambas sumatorias calculadas.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si el rango es [5..10], deberá imprimir 355.
- Si el rango es [3..8], deberá imprimir 199.
- Si el rango es [4..4], deberá imprimir 0.
- Si el rango es [7..11], deberá imprimir 415.
- Si el rango es [8..3], deberá imprimir 0.
- Si el rango es [14..19], deberá imprimir 1651.

2.2.4. La ecuación de la recta

Una recta se puede representar mediante una ecuación de la siguiente forma $y = mx + b$, en donde el valor de m corresponde a la pendiente y el valor de b corresponde al punto de intersección en la ordenada. Se desea determinar si dado determinado punto $P(x, y)$, este punto pertenece o no a una recta.

Sugerencia

- Lea en una variable el valor de la pendiente m y en otra variable el valor del punto de intersección en la ordenada b.
- Lea en una variable el valor de la abscisa x y en otra variable la ordenada y.
- Utilizando la ecuación de la recta $mx + b$ calcule el valor que debería tener la ordenada para x. Almacene este valor en una variable denominada y_recta.
- Compare el valor y con el valor de y_recta, si son iguales imprima el texto `El punto forma parte de la recta`. En caso contrario imprima el texto `El punto no forma parte de la recta`.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $m = 5$, $b = 3$, los siguientes son puntos de la recta $(1, 8)$, $(4, 23)$, $(6, 33)$
- Si $m = -3$, $b = 0$, los siguientes son puntos de la recta $(-3, 9)$, $(1, -3)$, $(10, -30)$
- Si $m = -1$, $b = 10$, los siguientes son puntos de la recta $(0, 10)$, $(5, 5)$, $(10, 0)$

2.2.5. La ecuación de la circunferencia

Una circunferencia se puede describir por la ecuación $(x - a)^2 + (x - b)^2 = r^2$, donde el punto (a, b) representa el centro de la circunferencia y r representa el radio de la circunferencia. Se pide que dada una ecuación de una circunferencia y un punto (x, y) en un plano cartesiano, determine si el punto se encuentra dentro del círculo delimitado por la circunferencia descrita por la ecuación.

Sugerencia

- Lea el punto (a, b) que representa el centro.
- Lea el radio r.
- Lea el punto (x, y)
- Halle el valor de $(x - a)^2 + (x - b)^2$, si este valor es menor o igual que r^2 imprima el texto `El punto está dentro del círculo`. En caso contrario imprima el texto `El punto no está dentro del círculo`.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $a = 2$, $b = 3$, $r = 5$, $x = 4$ e $y = 6$, se deberá imprimir `El punto está dentro del círculo`.
- Si $a = 2$, $b = 3$, $r = 5$, $x = 9$ e $y = 3$, se deberá imprimir `El punto no está dentro del círculo`.
- Si $a = 8$, $b = 9$, $r = 3$, $x = 9$ e $y = 10$, se deberá imprimir `El punto está dentro del círculo`.
- Si $a = 8$, $b = 9$, $r = 3$, $x = 14$ e $y = 10$, se deberá imprimir `El punto no está dentro del círculo`.

2.2.6. Cálculo del Interés simple

Un egresado de la Pontificia Universidad Católica del Perú posee un capital C de 50.000 soles y desea depositarlos en un banco a un plazo n de 3 años. La tasa de interés simple i que le ofrecen es de 30 % anual. Si el egresado en cuestión desea tener al final un saldo S de por lo menos 90.000 soles, ¿Le conviene hacer el depósito en el banco?

Recordar que:

El saldo final S se calcula de la siguiente manera:

$$S = C(1 + n * i)$$

Donde:

- n está expresado en años.
- i es el interés anual, valor del porcentaje multiplicado por $0,01$.

Sugerencia

- Calcule el saldo final.
  - Si el saldo final es mayor que el monto que se desea obtener se concluye que la operación es conveniente.
  - Si el saldo final es menor o igual que el monto que se desea obtener se concluye que la operación no es conveniente.

Sugerencia

A pesar que este problema puede ser resuelto de forma particular, generalice su solución para:

- Leer el capital C.
- Leer el plazo n.
- Leer la tasa de interés simple anual i.
- Leer el saldo final deseado S.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $c = 50000$ soles, $n = 3$ años, $i = 30\,\%$ y el saldo final deseado $S = 90000$ soles, se debe imprimir `Conviene depositar en el banco`.
- Si $c = 100000$ soles, $n = 5$ años, $i = 32\,\%$ y el saldo final deseado $S = 350000$ soles, se debe imprimir `No conviene depositar en el banco`.
- Si $c = 70000$ soles, $n = 4$ años, $i = 25\,\%$ y el saldo final deseado $S = 135000$ soles, se debe imprimir `Conviene depositar en el banco`.

2.2.7. ¿Se mueve o no se mueve la caja?

Sobre un plano horizontal se tiene una caja que pesa 35 N. Se sabe además que se tiene una coeficiente de rozamiento estático de $\mu_s = 0.5$. Si se le aplica una fuerza de 10 N, ¿la caja se mueve?

Recordar que:

La fuerza de rozamiento estático máximo se calcula de la siguiente manera:

$$fs\_max = \mu_s \times N$$

Donde:

- μ_s es el coeficiente de rozamiento estático.
- N es la fuerza normal.

Sugerencia

- Calcule la fuerza de rozamiento estático máximo. Recuerde que la fuerza de rozamiento estático máximo equivale a la fuerza mínima para iniciar un movimiento.
- Si la fuerza aplicada es mayor que la fuerza de rozamiento estático máximo, la caja se moverá.
- Si la fuerza aplicada es menor o igual que la fuerza de rozamiento estático máximo, la caja no se moverá.

Sugerencia

A pesar que este problema puede ser resuelto de forma particular, generalice su solución para:

- Leer el peso de una caja en N.
- Leer el coeficiente de rozamiento estático μ_s.
- Leer la fuerza que se aplicará en N.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $peso = 35N$, $\mu_s = 0.5$ y la fuerza aplicada es $10N$, se debe imprimir `No se mueve la caja`.
- Si $peso = 35N$, $\mu_s = 0.5$ y la fuerza aplicada es $18N$, se debe imprimir `Sí se mueve la caja`.
- Si $peso = 50N$, $\mu_s = 0.3$ y la fuerza aplicada es $16N$, se debe imprimir `Sí se mueve la caja`.
- Si $peso = 25N$, $\mu_s = 0.8$ y la fuerza aplicada es $19N$, se debe imprimir `No se mueve la caja`.

2.2.8. ¿Cómo puedo saber si mi peso es normal?

El Índice de Masa Corporal (IMC) es una razón matemática que asocia la masa y la talla de un individuo. Se calcula de la siguiente manera:

$$IMC = \frac{masa}{estatura^2}$$

Donde la *masa* se expresa en kg y la *estatura* en m^2. De acuerdo a la Organización Mundial de la Salud, si el IMC se encuentra entre los valores de 18.50 y 24.90, se dice que el peso es normal. Dado el valor de la *masa* en kg y la *estatura* en m^2 determine si el peso es normal.

Sugerencia

- Lea en una variable el valor de la *masa*.
- Lea en una variable el valor de la *estatura*.
- Con los valores leídos calcule el IMC.
- Si el valor calculado se encuentra en el rango [18.50..24.90] imprima el texto `El peso es normal`. En caso contrario imprima el texto `El peso no es normal`.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $masa = 86\ kg$ y $estatura = 1.70\ m$, se deberá imprimir `El peso no es normal`.
- Si $masa = 71\ kg$ y $estatura = 1.70\ m$, se deberá imprimir `El peso es normal`.
- Si $masa = 75\ kg$ y $estatura = 1.65\ m$, se deberá imprimir `El peso no es normal`.
- Si $masa = 79\ kg$ y $estatura = 1.80\ m$, se deberá imprimir `El peso es normal`.

2.2.9. Funciones definida por tramos

Considerando la función $f(x)$ definida de la siguiente manera:

$$f(x) = \begin{cases} |x^3| & x \leq -1 \\ x^2 + 4x & -1 < x \leq 6 \\ \lfloor \dfrac{x^5 + 5x}{3} \rfloor & 6 < x \leq 14 \\ \log_5 x & x > 14 \end{cases}$$

Se solicita que dado un valor de $x \in \mathbb{R}$ determine el valor de la función $f(x)$.

Sugerencia en PSeInt

- En PseInt el valor absoluto de un número real se puede obtener usando la función `abs`.
- En PseInt la potencia de un número se puede obtener usando el operador `^`.
- En PseInt no existe la función piso.
- En PseInt no existe una función para calcular el logaritmo en cualquier base, para esto debe usar el teorema de cambio de base.

Sugerencia en C

- En C el valor absoluto de un número real se puede obtener usando la función `fabs` cuya declaración se encuentra en el archivo de cabecera `math.h`.
- En C la potencia de un número se puede obtener usando la función `pow` cuya declaración se encuentra en el archivo de cabecera `math`.
- En C la función piso de un número se puede obtener usando la función `floor` cuya declaración se encuentra en el archivo de cabecera `math.h`.
- En C no existe una función para calcular el logaritmo en cualquier base, para esto debe usar el teorema de cambio de base.

2.2.10. Categorización del Índice de Masa Corporal

El sobrepeso y la obesidad se definen como una acumulación anormal o excesiva de grasa que puede ser perjudicial para la salud. El índice de masa corporal (IMC) es un indicador simple de la relación entre el peso y la talla que se utiliza frecuentemente para identificar el sobrepeso y la obesidad en los adultos. Se calcula dividiendo el peso de una persona en kilos por el cuadrado de su talla en metros $(kg/m2)$[2]. La **OMS** clasifica el estado nutricional utilizando el IMC conforme la tabla 2.2.

Tabla 2.2: Categorías de Índice de Masa Corporal

Clasificación	**IMC**(Kg/m^2)
Bajo peso	< 18.50
Normal	$[18.50 - 24.99]$
Preobeso	$[25.00 - 29.99]$
Obesidad leve	$[30.00 - 34.99]$
Obesidad media	$[35.00 - 39.99]$
Obesidad mórbida	≥ 40.00

Se solicita que dada el peso de una persona en *kg* y su talla en *metros*, determine el estado nutricional de dicha persona.

[2] http://www.who.int/es/news-room/fact-sheets/detail/obesity-and-overweight

2.2.11. La Media Aritmética-Geométrica

La media aritmética-geométrica (MAG) es una serie muy usada en las matemáticas para acelerar algoritmos que realizan cálculos de funciones exponenciales y trigonométricas. También ha sido utilizada para calcular constantes matemáticas como π.

Recordar que:

En matemáticas, la media aritmética-geométrica (MAG) de dos números reales positivos x e y es definida como sigue:

Primero se definen los valores a_o y g_0 como $a_0 = x$ y $g_0 = y$. Luego, en base a estos valores, se definen las secuencias interdependientes (a_n) y (g_n) de la siguiente manera:

$$a_{n+1} = \frac{a_n + g_n}{2}$$

$$g_{n+1} = \sqrt{a_n \times g_n}$$

Una propiedad interesante de esta serie es que dado un número real $r \geq 0$, se cumple que $MAG(rx, ry) = rMAG(x, y)$.

Se le pide que dado los números x, y y r, verifique haciendo uso del computador, si esta propiedad se cumple.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $x = 33$, $y = 12$ y $r = 3$ entonces se debe imprimir $a_n = 67.5$, $g_n = 59.69$, `Se cumple propiedad.`
- Si $x = 10$, $y = 5$ y $r = 5$ entonces se debe imprimir $a_n = 37.5$, $g_n = 35.35$, `Se cumple propiedad.`
- Si $x = 4$, $y = 21$ y $r = 9$ entonces se debe imprimir $a_n = 112.5$, $g_n = 82.48$, `Se cumple propiedad.`

2.2.12. Calculadora de operaciones vectoriales

Dado dos vectores en el plano cartesiano $u = (u_x, u_y)$ y $v = (v_x, v_y)$ y un caracter que representa una operación vectorial ($+$ que representa a la operación de suma de vectores, $-$ que representa a la operación de diferencia de vectores, $\cdot$ que representa a la operación de producto escalar), se solicita que retorne el resultado de aplicar la operación vectorial a los valores dados. En caso que la operación ingresada no corresponda con las antes mencionadas, deberá emitirse un mensaje de advertencia y no realizar las operaciones.

Recordar que:

Dado dos vectores en el plano cartesiano $u = (u_x, u_y)$ y $v = (v_x, v_y)$. Se definen las siguientes operaciones:

- suma vectorial: $u + v = (u_x + v_x, u_y + v_y)$.
- diferencia vectorial: $u - v = (u_x - v_x, u_y - v_y)$.
- producto escalar: $u.v = u_x \times v_x + u_y \times v_y$.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $u = (3, 0)$, $v = (5, 5)$ y $operación = +$ se debe imprimir `El resultado es u+v=(8,5)`.
- Si $u = (3, 0)$, $v = (5, 5)$ y $operación = -$ se debe imprimir `El resultado es u-v=(-2,-5)`.
- Si $u = (3, 0)$, $v = (5, 5)$ y $operación = .$ se debe imprimir `El resultado es u.v=15`.

2.2.13. Distancia más cercana

Dados 3 puntos A, B, C en el plano cartesiano ($P_A(x_A, y_A)$, $P_B(x_B, y_B)$, $P_C(x_C, y_C)$), se le solicita que imprima el par de puntos que contiene la menor distancia. Asuma que las distancias del punto entre todos los puntos siempre serán diferentes.

Recuerdar que:

Recuerde que la distancia euclidiana d se calcula de la siguiente manera:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $P_A = (3, 5)$, $P_B = (4, 2)$ y $P_C = (7, 7)$, debe retornar `La menor distancia se encuentra entre el punto A y B`.
- Si $P_A = (1, 3)$, $P_B = (6, 7)$ y $P_C = (4, 5)$, debe retornar `La menor distancia se encuentra entre el punto B y C`.
- Si $P_A = (9, 8)$, $P_B = (5, 7)$ y $P_C = (7, 8)$, debe retornar `La menor distancia se encuentra entre el punto A y C`.

Sugerencia

Realice los cambios necesarios para que ahora imprima los puntos que conforman la distancia más lejana. Una vez hecho los cambios, utilice los siguientes datos para probar su solución.

- Si $P_A = (3, 5)$, $P_B = (4, 2)$ y $P_C = (7, 7)$, debe retornar `La mayor distancia se encuentra entre el punto B y C`.
- Si $P_A = (1, 3)$, $P_B = (6, 7)$ y $P_C = (4, 5)$, debe retornar `La mayor distancia se encuentra entre el punto A y B`.
- Si $P_A = (9, 8)$, $P_B = (5, 7)$ y $P_C = (7, 8)$, debe retornar `La mayor distancia se encuentra entre el punto A y B`.

2.2.14. Calculadora de números complejos

Dado dos números complejos representados en forma binomial (i.e., $z = a + bi$) y un caracter que representa una operación aritmética ($+$, $-$, $*$ y $/$), se solicita que retorne el resultado de aplicar la operación en los dos números complejos dados.

Recordar que:

Dado dos números complejos $a + bi$ y $c + di$. Se definen las siguientes operaciones:

- suma: $(a + bi) + (c + di) = (a + c) + (b + d)i$
- resta: $(a + bi) - (c + di) = (a - c) + (b - d)i$
- multiplicación: $(a + bi) * (c + di) = (ac - bd) + (ad + bc)i$
- división: $\frac{a + bi}{c + di} = \frac{ac + bd}{c^2 + d^2} + (\frac{bc - ad}{c^2 + d^2})i$

Sugerencia

- Utilice 2 variables para leer un número complejo, una para representar la parte real y otra para representar la parte imaginaria. Por ejemplo para el primer número podría utilizar las variables a y b y para el segundo número las variables c y d.
- Verifique que no exista división entre 0.
- Si el resultado de una operación retorna la parte imaginaria negativa, entonces deberá imprimirse la parte imaginaria con el signo negativo (e.g., $a - bi$), en caso contrario deberá imprimirse el signo positivo (e.g., $a + bi$).

2.2.15. Tipos de rectas

La pendiente de una recta permite identificar el tipo de inclinación que posee la recta. Si la pendiente es positiva, entonces la recta será `ascendente`. Si la pendiente es negativa, la recta será `descendente`. Si la pendiente tiene el valor de cero, la recta será `horizontal`. Si la pendiente no está definida, entonces la recta será `vertical`.

Se le pide que dados dos puntos $P1(x_1, y_1)$ y $P2(x_2, y_2)$ que pertenecen a una recta, determine e imprima la inclinación de la recta.

Recuerdar que:

Si $P1(x_1, y_1)$ y $P2(x_2, y_2)$ dos dos puntos de una recta, entonces la pendiente m se calcula de la siguiente manera:

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $P1(1, 4)$ y $P2(3, 9)$, entonces se debe imprimir $m = 2.5$ `recta ascendente`.
- Si $P1(0, 1)$ y $P2(0, 7)$, entonces se debe imprimir $m = indefinida$ `recta vertical`.
- Si $P1(-1, -4)$ y $P2(3, 2)$, entonces se debe imprimir $m = 1.5$ `recta ascendente`.
- Si $P1(2, 0)$ y $P2(4, 0)$, entonces se debe imprimir $m = 0$ `recta horizontal`.
- Si $P1(3, 5)$ y $P2(2, 8)$, entonces se debe imprimir $m = -3$ `recta descendente`.

2.3. Nivel Avanzado

2.3.1. Distancia más cercana

Dados 3 puntos A, B, C en el plano cartesiano ($P_A(x_A, y_A)$, $P_B(x_B, y_B)$, $P_C(x_C, y_C)$) y un punto X ($P_X(x_X, y_X)$), se le solicita que imprima el punto que se encuentra más cercano a el punto X. Asuma que las distancias del punto X a los demás puntos serán siempre diferentes.

Recuerdar que:

Recuerde que la distancia euclidiana d se calcula de la siguiente manera:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Casos de prueba

- Si $P_A = (3, 5)$, $P_B = (4, 2)$, $P_C = (7, 7)$ y $P_X = (4, 4)$, se imprime `El punto más cercano es A.`
- Si $P_A = (3, 5)$, $P_B = (4, 2)$, $P_C = (7, 7)$ y $P_X = (2, 1)$, se imprime `El punto más cercano es B.`
- Si $P_A = (3, 5)$, $P_B = (4, 2)$, $P_C = (7, 7)$ y $P_X = (8, 7)$, se imprime `El punto más cercano es C.`

Variación al problema

Realice los cambios necesarios en el programa para que ahora imprima el punto más lejano al punto X. Una vez hecho los cambios, utilice los siguientes datos para probar su solución.

- Si $P_A = (3, 5)$, $P_B = (4, 2)$, $P_C = (7, 7)$ y $P_X = (11, 3)$, se imprime `El punto más lejano es A.`
- Si $P_A = (3, 5)$, $P_B = (4, 2)$, $P_C = (7, 7)$ y $P_X = (2, 1)$, se imprime `El punto más lejano es C.`
- Si $P_A = (3, 5)$, $P_B = (4, 2)$, $P_C = (7, 7)$ y $P_X = (8, 7)$, se imprime `El punto más lejano es B.`

2.3.2. Clasificación de un triángulo según sus lados

Dados 3 números reales que representan los lados de un triángulo, se le solicita que luego de verificar si los 3 lados forman un triángulo, imprima el tipo de triángulo según sus lados.

Recordar que:

La clasificación de triángulos según sus lados es como sigue:

Equilátero si sus 3 lados son iguales.

Isósceles si 2 de sus lados, y solo 2, son iguales.

Escaleno si los 3 lados son diferentes.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $l1 = 3$, $l2 = 4$ y $l3 = 5$, deberá imprimirse `Triángulo Escaleno`.
- Si $l1 = 3$, $l2 = 3$ y $l3 = 2$, deberá imprimirse `Triángulo Isósceles`.
- Si $l1 = 3$, $l2 = 2$ y $l3 = 3$, deberá imprimirse `Triángulo Isósceles`.
- Si $l1 = 2$, $l2 = 3$ y $l3 = 3$, deberá imprimirse `Triángulo Isósceles`.
- Si $l1 = 10.5$, $l2 = 10.5$ y $l3 = 10.5$, deberá imprimirse `Triángulo Equilátero`.
- Si $l1 = 1$, $l2 = 10$ y $l3 = 1$, deberá imprimirse `Los lados ingresados no forman un triángulo`.

2.3.3. Área de un triángulo isósceles

Dados 3 lados que forman un triángulo isósceles, se le solicita que determine el área de dicho triángulo. Asuma que los 3 lados forman un triángulo. Asuma que siempre tendrá 2 lados iguales y uno diferente. Además el usuario podrá ingresar los lados en cualquier orden.

Recordar que:

El área de un triángulo isóceles se puede calcular usando la siguiente fórmula:

$$area = \frac{b}{4} \times \sqrt{4a^2 - b^2}$$

Donde:

- a es uno de los lados iguales.
- b es el lado diferente.

Restricciones

- No podrá aplicar la fórmula de Herón.

Sugerencia

- Halle primero el lado que se repite y asígnelo a la variable a.
- Halle luego el lado que no se repite y asígnelo a la variable b.
- Aplique la fórmula para calcular el área del triángulo isósceles.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $l1 = 3$, $l2 = 3$, $l3 = 2$, entonces el área debe ser ≈ 2.828427125.
- Si $l1 = 3$, $l2 = 2$, $l3 = 3$, entonces el área debe ser ≈ 2.828427125.
- Si $l1 = 2$, $l2 = 3$, $l3 = 3$, entonces el área debe ser ≈ 2.828427125.
- Si $l1 = 5$, $l2 = 4$, $l3 = 5$, entonces el área debe ser ≈ 9.16515139.

2.3.4. Operaciones con fracciones

Una fracción se puede representar usando 2 variables, una para el numerador y otra para el denominador. Se solicita que lea dos fracciones y una operación (`+`, `-`, `*`, `/`) y retorne el resultado de aplicar la operación a las dos fracciones.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si la fracción 1 es $\frac{1}{2}$, la fracción 2 es $\frac{1}{3}$ y la operación es $+$, se deberá retornar $\frac{5}{6}$
- Si la fracción 1 es $\frac{1}{2}$, la fracción 2 es $\frac{1}{3}$ y la operación es $-$, se deberá retornar $\frac{1}{6}$
- Si la fracción 1 es $\frac{2}{7}$, la fracción 2 es $\frac{5}{3}$ y la operación es $*$, se deberá retornar $\frac{10}{21}$
- Si la fracción 1 es $\frac{2}{7}$, la fracción 2 es $\frac{5}{3}$ y la operación es $/$, se deberá retornar $\frac{6}{35}$

2.3.5. Los números Armstrong

Un número Armstrong, también llamado número narcisista, es todo aquel número que es igual a la suma de cada uno de sus dígitos elevado al número total de dígitos.

A continuación siguen algunos ejemplos de números Armstrong:

- $371 = 3^3 + 7^3 + 1^3$. Total de dígitos 3.
- $8208 = 8^4 + 2^4 + 0^4 + 8^4$. Total de dígitos 4.
- $4210818 = 4^7 + 2^7 + 1^7 + 0^7 + 8^7 + 1^7 + 8^7$. Total de dígitos 7.

Dado un número n de exactamente 3 cifras, determine si dicho número es Armstrong. Si el número no tiene exactamente 3 cifras, no realice ningún procesamiento.

Sugerencia

- Para determinar si un número tiene exactamente 3 cifras, simplemente verifique que sea mayor o igual que 100 y menor o igual que 999.
- Para obtener las cifras de un número n de 3 cifras siga el siguiente algoritmo:
  - Obtenga el dígito de la unidades utilizando la operación de módulo del número n entre 10 (si $n = 153$, `153%10=3`).
  - Actualice el número n con el valor que resulta de dividir n entre 10 (si $n = 153$, `n=153/10=15`).
  - Obtenga el dígito de la decenas utilizando la operación de módulo del nuevo número n entre 10 (si $n = 15$, `15%10=5`).
  - Actualice el número n con el valor que resulta de dividir n entre 10 (si $n = 15$, `n=15/10=1`).
  - El dígito de las centenas estará en la variable n.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $n = 153$ se deberá imprimir `El número es Armstrong` $(153 = 1^3 + 5^3 + 3^3)$.
- Si $n = 271$ se deberá imprimir `El número no es Armstrong` $(352 = 2^3 + 7^3 + 1^3)$.
- Si $n = 370$ se deberá imprimir `El número es Armstrong` $(370 = 3^3 + 7^3 + 1^0)$.
- Si $n = 435$ se deberá imprimir `El número no es Armstrong` $(216 = 4^3 + 3^3 + 5^3)$.
- Si $n = 371$ se deberá imprimir `El número es Armstrong` $(371 = 3^3 + 7^3 + 1^3)$.

2.3.6. Números palíndromos

Un número natural es un palíndromo si se lee igual de izquierda a derecha y de derecha a izquierda. Son ejemplos de números palíndromos 1, 33, 141, 45654, 13531. Una forma para calcular un número palíndromo, pero que no siempre funciona, es:

- Seleccionar un número n.
- Invertir el número n
- Sumar al número invertido el valor de n
- El resultado de la suma es un candidato para ser palíndromo.

Este algoritmo algunas veces obtiene un número palíndromo. Por ejemplo:

```
n=13
inverso de n=31
suma = 13 + 31 = 44

n=65
inverso de n=56
suma 65 + 56 = 121
```

Dado un número n de exactamente 2 cifras sin incluir el 0, determine si con el algoritmo antes presentado se puede encontrar un número palíndromo. En caso de encontrarlo deberá imprimir dicho número palíndromo. Si no lo encuentra deberá imprimirse el mensaje `No se encontró el palíndromo`.

Sugerencia

Para invertir un número n de 2 cifras siga el siguiente algoritmo:

- Obtenga el dígito de la unidades utilizando la operación de módulo del número n entre 10 (si $n = 13$, `13%10=3`).
- Obtenga el dígito de la decenas utilizando la operación de división entera del número n entre 10 (si $n = 13$, `13/10=1`).
- Multiplique el dígito de las unidades por 10 y súmele el dígito de las decenas (si $n = 13$, `3*10+1=31`)
- El resultado de la suma será el número invertido (si $n = 13$, el invertido es 31).

Sugerencia

Para invertir un número n de 3 cifras siga el siguiente algoritmo:

- Obtenga el dígito de la unidades utilizando la operación de módulo del número n entre 100 (si $n = 156$, `156 %100=6`).
- Obtenga un nuevo número m que resulta de dividir n entre 10 (si $n = 156$, `m=156/10=15`).
- Invierta el número m que ahora tiene dos cifras (si $m = 15$ el invertido será 51). Como el número tiene 2 cifras puede utilizar el algoritmo mencionado previamente.
- Multiplique el dígito de la unidades por 100 y súmele el número m (`6*100+51=651`)
- El resultado de la suma será el número invertido (si $n = 156$, el invertido es 651).

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $n = 15$ se deberá imprimir `Se encontró el palíndromo 66`.
- Si $n = 64$ se deberá imprimir `No se encontró el palíndromo`.
- Si $n = 25$ se deberá imprimir `Se encontró el palíndromo 77`.
- Si $n = 78$ se deberá imprimir `No se encontró el palíndromo`.

2.3.7. Clasificación de un paralelogramo según sus lados

Un cuadrilátero se define en la geometría euclidiana como un polígono de 4 lados. Dependiendo de la cantidad de lados paralelos, la longitud de sus lados y el valor de sus ángulos internos, los cuadriláteros se pueden clasificar en distintas categorias (ver figura 2.1).

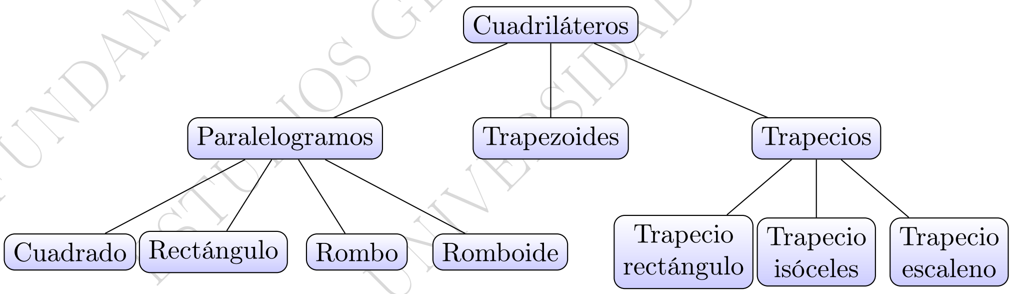

Figura 2.1: Clasificación de cuadriláteros

Los `Cuadriláteros` se pueden clasificar en `Paralelogramos`, `Trapezoides` y `Trapecios`. Los `Paralelogramos` poseen exactamente 4 lados paralelos, los `Trapecios` poseen exactamente 2 lados paralelos y los `Trapezoides` no poseen lados paralelos.

Dentro de los `Paralelogramos`, el `Cuadrilátero` se puede clasificar como `Cuadrado`, `Rectángulo`, `Rombo` y `Romboide`. El `Cuadrado` se caracterizar por tener sus 4 lados iguales y sus 4 ángulos rectos. El `Rectángulo` también tiene sus 4 ángulos rectos pero tiene 2 pares de lados iguales. El `Rombo` tiene tiene 2 pares de ángulos iguales pero en este caso los 4 lados deben ser iguales. El `Romboide` por su lado se caracteriza por tener también 2 pares de ángulos iguales con 2 pares de lados iguales.

Dentro de los `Trapecios`, el `Cuadrilátero` se puede clasificar como `Trapecio rectángulo`, `Trapecio isóceles` y `Trapecio escaleno`. El `Trapecio rectángulo` se caracteriza por tener 2 ángulos rectos. El `Trapecio isóceles` se caracterizar por tener 2 ángulos internos iguales. Por otro lado el `Trapecio escaleno` no posee ningún ángulo recto y además todos sus ángulos son diferentes.

Recordar que:

Una propiedad de los cuadriláteros es que la suma de los ángulos internos suma $360°$ o 2π radianes.

Se pide que dado 4 lados de un cuadrilatero, determine el posible paralelogramo en el cual se podría clasificar.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $l1 = 4, l2 = 4, l3 = 4, l4 = 4$, entonces se debe imprimir `Posible Cuadrado o Rombo`.
- Si $l1 = 4, l2 = 6, l3 = 6, l4 = 4$, entonces se debe imprimir `Posible Rectángulo o Romboide`.
- Si $l1 = 6, l2 = 4, l3 = 6, l4 = 4$, entonces se debe imprimir `Posible Rectángulo o Romboide`.
- Si $l1 = 3, l2 = 3, l3 = 5, l4 = 5$, entonces se debe imprimir `Posible Rectángulo o Romboide`.

Sugerencia

Realice las siguientes modificaciones al problema:

- De forma análoga, dados los ángulos internos del cuadrilátero, determine el posible tipo de trapecio que corresponde con el cuadrilátero.
- Usando ahora tanto los lados del cuadrilátero así como sus ángulos internos, determine el correcto tipo de paralelogramo.
- Dados la longitud de cada uno de los 4 lados, el valor en grados sexagesimales de cada ángulo interno así como la cantidad de lados paralelos que existen en un cuadrilátero y determine qué tipo de cuadrilátero es.

2.3.8. Comparación de fechas

Dados dos fechas $f_1 = dd_1/mm_1/aaaa_1$ y $f_2 = dd_2/mm_2/aaaa_2$, donde dd corresponde al día, mm corresponde al mes y $aaaa$ corresponde el año. Se le pida que las compare y retorne 0 en caso las fechas sean iguales. Si la fecha f_1 es mayor que la fecha f_2 retornará un número positivo. Si la fecha f_1 es menor que la fecha f_2 retornará un número negativo. En caso las fechas sean diferentes, la magnitud del número corresponderá a la diferencia que existen entre ambas fechas.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $f_1 = 29/11/1976$ y $f_2 = 29/11/1976$ se debe imprimir `La comparación retorna 0`.
- Si $f_1 = 27/04/2004$ y $f_2 = 26/04/2004$ se debe imprimir `La comparación retorna 1`.
- Si $f_1 = 23/07/2005$ y $f_2 = 24/07/2005$ se debe imprimir `La comparación retorna -1`.
- Si $f_1 = 10/01/2017$ y $f_2 = 24/04/2019$ se debe imprimir `La comparación retorna -834`.

Recuerdar que:

Recuerde que la cantidad de días transcurridos hasta la fecha se puede calcular según la siguiente fórmula:

$$d = ((aaaa - 1) * 365 + \frac{aaaa - 1}{4} - (3 * \frac{\frac{aaaa-1}{100} + 1}{4}) + dm + d$$

Donde:

- $aaaa$ corresponde al año.
- dm corresponde a los corresponden a los días transcurridos hasta el mes anterior.
- d corresponde al día.
- La división corresponde a la división entera.

Sugerencia

Para obtener la cantidad de días transcurridos entre dos fechas:

- Calcule la cantidad de días transcurridos a la fecha 1.
- Calcule la cantidad de días transcurridos a la fecha 2.
- Obtenga la diferencia entre los días mediante usando la operación de diferencia con la cantidades calculadas en los pasos anteriores.

2.3.9. Conversión de coordenadas rectangulares a polares

Dado un punto $P(x, y)$ en un plano cartesiano, se pide representar el punto en coordenadas polares.

Recordar que:

En las coordenadas polares un punto está definido por dos componentes, la coordenada radial r y la coordenada angular θ. Dado un punto $P(x, y)$, se tiene que:
$r = \sqrt{x^2 + y^2}$

$$\theta = \begin{cases} arctan(\frac{y}{x}) & x > 0 \wedge y \geq 0 \\ \frac{\pi}{2} & x = 0 \wedge y > 0 \\ arctan(\frac{y}{x}) + \pi & x < 0 \\ \frac{3\pi}{2} & x = 0 \wedge y < 0 \\ arctan(\frac{y}{x}) + 2\pi & x > 0 \wedge y < 0 \end{cases}$$

Sugerencia para PSeInt

- En PSeIntel arco tangente de un número real se puede obtener usando la función `atan`. El resultado de esta función se retorna en radianes.

Sugerencia para C

- En C el arco tangente de un número real se puede obtener usando la función `atan` cuya declaración se encuentra en el archivo de cabecera `math.h`. El resultado de esta función se retorna en radianes.

2.3.10. Suma de n números naturales

Dado un número n que representa una cantidad de números naturales y un número y que representa un exponente, se solicita que retorne la suma de las potencias de y de los n primeros números naturales. Se sabe que $n > 0$ e $y \in [1..3]$.

Recordar que:

La suma de los n primeros números naturales se puede calcular mediante la siguiente serie notable:

$$\sum_{i=1}^{n} i = 1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$$

Recordar que:

La suma de los cuadrados de los n primeros números naturales se puede calcular mediante la siguiente serie notable:

$$\sum_{i=1}^{n} i^2 = 1^2 + 2^2 + 3^2 + \cdots + n^2 + = \frac{n(n+1)(2n+1)}{6}$$

Recordar que:

La suma de los cubos de los n primeros números naturales se puede calcular mediante la siguiente serie notable:

$$\sum_{i=1}^{n} i^3 = 1^3 + 2^3 + 3^3 + 4^3 \cdots + n^3 = \frac{n^2(n+1)^2}{4}$$

2.3.11. Manipulación de número enteros

Dado un número $n \in \mathbb{N}$ y un caracter o que representa un caracter, se desea que se retorne la inversa del número n cuando el caracter o es igual a `'I'` y retorne la suma de los dígitos elevado al cubo cuando el caracter o es igual a `'A'`. La cantidad de dígitos del número deberá ser de exactamente 3 caracteres. Si el caracter no es ni `'A'` ni `'I'`, deberá retornar el mensaje `Opción inválida`. Si la cantidad de dígitos no es la indicada, deberá retornar el mensaje `Debe ingresar un número de 3 dígitos`.

El siguiente ejemplo muestra un caso de ejecución cuando se desea invertir un número.

```
Ingrese número n: -153
Ingrese opción o: I

El número invertido es: -351
```

El siguiente ejemplo muestra un caso de ejecución cuando se desea obtener la suma de los dígitos al cubo del número.

```
Ingrese número n: 121
Ingrese opción o: A

La suma de dígitos al cubo es: 10
```

El siguiente ejemplo muestra un caso de ejecución cuando la opción ingresada es incorrecta.

```
Ingrese número n: 146
Ingrese opción o: W

Opción inválida
```

El siguiente ejemplo muestra un caso de ejecución cuando la cantidad de dígitos del número no es la adecuada.

```
Ingrese número n: 4578
Ingrese opción o: A

Debe ingresar un número de 3 dígitos
```

2.3.12. Cálculo de las raíces ecuaciones cuadráticas con una variable

Se pide que dada una ecuación cuadrática con una variable, se realice el cálculo de las raíces y presente las raíces de dicha ecuación. En caso que el ecuación tenga raíces complejas, deberá solicitar al usuario si desea expresar las raíces complejas usando la representación binómica o la representación polar.

Recordar que:

La representación binómica de un número complejo es de la forma $x + yi$
donde:

- x es la parte real
- y es la parte imaginaria

Recordar que:

La representación polar de un número complejo es de la forma $|z|(cos(\theta) + isen(\theta))$
Si se tiene el número de la forma $x + yi$, se puede calcular:

- $|z| = \sqrt{x^2 + y^2}$
- $cos(\theta) = \frac{x}{\sqrt{x^2+y^2}}$
- $sen(\theta) = \frac{y}{\sqrt{x^2+y^2}}$

Recordar que:

Para verificar qué tipo de representación desea visualizar el usuario, utilice una opción para poder realizar la decisión utilizando una selectiva simple, de forma similar al problema propuesto 2.3.11.

2.3.13. Ecuación cúbica de una variable (adaptado del laboratorio 3 2020-1)

La ecuación cúbica de una variable es una expresión de la siguiente forma:

$$ax^3 + bx^2 + cx + d = 0$$

Donde los coeficientes (a, b, c y d) son números reales y además se debe cumplir necesariamente que a es diferente de cero. Se sabe además que cuando el discriminante es mayor que cero, la ecuación tiene 3 raíces reales distintas. El discriminante de una ecuación cúbica es $18abcd - 4b^3d + b^2c^2 - 4ac^3 - 27a^2d^2$

El programa deberá realizar lo siguiente:

- Leer los coeficientes.
- Verificar si la ecuación es cúbica.
- Determinar si la ecuación posee 3 raíces reales distintas. Cuando la ecuación se verifique como cúbica y el discriminantes sea mayor que cero, debe emitirse el siguiente mensaje `La ecuación tiene 3 raíces reales distintas`. En caso contrario debe emitirse el siguiente mensaje `La ecuación no tiene 3 raíces reales distintas o no es cúbica`.

Casos de prueba para verificar la solución

Use los siguientes casos para verificar si la solución está correcta.

a	b	c	d	Impresión
0	5	3	3	La ecuación no tiene 3 raíces reales distintas o no es cúbica.
5	1	2	16	La ecuación no tiene 3 raíces reales distintas o no es cúbica.
2	-5	-9	18	La ecuación tiene 3 raíces reales distintas.
2	-7	7	-2	La ecuación tiene 3 raíces reales distintas.

2.3.14. Áreas de figuras planas (adaptado del laboratorio 3 2020-1)

Dado los 3 lados de un triángulo escaleno (a, b y c) y el lado de un héxagono regular (l). Elabore un programa en lenguaje C que permite determinar cuál de las dos figuras geométricas posee el área mayor. Asuma que los lados ingresados forman un triángulo.

El programa deberá realizar lo siguiente:

- Leer los lados del triángulo.
- Leer el lado del hexágono.
- Calcular el área del triángulo.
- Calcular el área del hexágono.
- Determinar la mayor área. Deberá imprimir que figura posee la mayor área.

Casos de prueba para verificar la solución

Use los siguientes casos para verificar si la solución está correcta.

a	b	c	l	Impresión
3	4	5	1.5	El triángulo posee mayor área.
2.24	2.83	1	3	El hexágono posee mayor área.
11	11	7.4	3.2	El triángulo posee mayor área.
4	5	9	7.8	El hexágono posee mayor área.

Recordar que:

Según la fórmula de Herón el área del triángulo se calcula de la siguiente manera:

$$area = \sqrt{s(s-a)(s-b)(s-c)}$$

donde:

- $s = \dfrac{a+b+c}{2}$.
- a, b y c son los lados del triángulo.

Recordar que:

El área de un hexágono regular se calcula de la siguiente manera:

$$area = \frac{3l^2}{2tan(30°)}$$

Recordar que:

$360° = 2\pi$ radianes

Sugerencia para lenguaje C

La función `tan` retorna la tangente del ángulo pasado como parámetro. El ángulo debe estar expresado en radianes. Esta función se encuentra definida en el archivo de cabecera `math.h`

Variación al problema

Desarrolle el programa para un **triángulo equilátero y un decágono regular**.
El área de un triángulo equilátero se puede calcular usando la siguiente fórmula:

$$area = \frac{lado^2\sqrt{3}}{4}$$

No debe usar la fórmula de Herón para calcular el área del triángulo equilátero.
El área de un decágono regular se calcula de la siguiente manera:

$$area = \frac{10l^2}{4tan(18°)}$$

Use los siguientes casos de prueba para verificar si la solución está correcta.

a	l	Impresión
8	1.5	El triángulo posee mayor área.
1	3	El decágono posee mayor área.
7.4	1.2	El triángulo posee mayor área.
12	7.8	El decágono posee mayor área.

Variación al problema

Desarrolle el programa para un **triángulo isósceles y un heptágono regular**.
El área de un triángulo isósceles se puede calcular usando la siguiente fórmula:

$$area = \frac{b}{4} \times \sqrt{4a^2 - b^2}$$

Donde:

- a es uno de los lados iguales.
- b es el lado diferente.

No debe usar la fórmula de Herón para calcular el área del triángulo isósceles.
El área de un heptágono regular se calcula de la siguiente manera:

$$area = \frac{7l^2}{4tan(25.71°)}$$

Use los siguientes casos de prueba para verificar si la solución está correcta.

a	b	l	Impresión
3	4	0.5	El triángulo posee mayor área.
2.24	2.83	3	El heptágono posee mayor área.
11	7.4	2.6	El triángulo posee mayor área.
4	5	7.8	El heptágono posee mayor área.

2.3.15. Ecuación de segundo grado (adaptado del laboratorio 3 2020-1)

La ecuación general de segundo grado es una expresión con dos variables x e y de la siguiente forma:

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$$

Donde los coeficientes (A, B, C, D, E y F) son números reales y además se debe cumplir necesariamente que al menos uno de los valores de los coeficientes A, B o C es diferente de cero. Se sabe además que cuando $AC - \frac{B^2}{4} < 0$ se dice que la ecuación es de tipo hiperbólica.

El programa deberá realizar lo siguiente::

- Leer los coeficientes.
- Verificar si la ecuación es general de segundo grado.
- Determinar si la ecuación es hiperbólica. Cuando la ecuación sea general de segundo grado y sea del tipo hiperbólica, debe emitirse el siguiente mensaje `La ecuación es de tipo hiperbólica`. En caso contrario debe emitirse el siguiente mensaje `La ecuación no es de segundo grado o no es de tipo hiperbólica`.

Casos de prueba para verificar la solución

Use los siguientes casos para verificar si la solución está correcta.

A	B	C	D	E	F	Impresión
0	0	0	3.3	1.2	4.5	La ecuación no es de segundo grado o no es de tipo hiperbólica.
1	0	1	3.6	2.7	5.7	La ecuación no es de segundo grado o no es de tipo hiperbólica.
1.3	4.5	2.5	3.5	1.6	4.5	La ecuación es de tipo hiperbólica.
-2.7	-5.4	-2.4	2.6	5.6	1.2	La ecuación es de tipo hiperbólica.

Variación al problema

Desarrolle el programa para verificar si la ecuación es de tipo **parabólica**, cuando $AC - \frac{B^2}{4} = 0$.
Use los siguientes casos de prueba para verificar si la solución está correcta. Solo debe leer los coeficientes A, B y C.

A	B	C	Impresión
0	0	0	La ecuación no es de segundo grado o no es de tipo parabólica.
1	0	1	La ecuación no es de segundo grado o no es de tipo parabólica.
1	2.5	1.5625	La ecuación es de tipo parabólica.
4	8	4	La ecuación es de tipo parabólica.

Variación al problema

Desarrolle el programa para verificar si la ecuación **no es de tipo elíptica**. Una ecuación de segundo grado es de tipo elíptica cuando $AC - \frac{B^2}{4} > 0$.
Use los siguientes casos de prueba para verificar si la solución está correcta. Solo debe leer los coeficientes A, B y C.

A	B	C	Impresión
0	0	0	La ecuación no es de segundo grado o es de tipo elíptica.
1.5	2.5	4.5	La ecuación no es de segundo grado o es de tipo elíptica.
2.3	5.4	1.6	La ecuación no es de tipo elíptica.
5.1	7.4	0.4	La ecuación no es de tipo elíptica

Variación al problema

Desarrolle el programa para verificar si la ecuación **no es de tipo circunferencia**, cuando $A = C$.
Use los siguientes casos de prueba para verificar si la solución está correcta.

A	B	C	D	E	F	Impresión
0	0	0	3.3	1.2	4.5	La ecuación no es de segundo grado o es una circunferencia
1	0	1	3.6	2.7	5.7	La ecuación no es de segundo grado o es una circunferencia
1.3	4.5	2.5	3.5	1.6	4.5	La ecuación no es una circunferencia.
-2.7	-5.4	-2.4	2.6	5.6	1.2	La ecuación no es una circunferencia.

2.3.16. Ecuación de una paraboloide (adaptado del laboratorio 3 2020-1)

Un paraboloide se puede expresar a través de una ecuación cuadrática definida en un espacio tridimensional, por lo que los puntos se definen en base a tres coordenadas x, y y z. Una de las formas canónicas del paraboloide es:

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 - z = 0$$

En donde los coeficientes a y b son números reales diferentes de cero.

El programa deberá realizar lo siguiente:

- Leer los coeficientes de la ecuación (a y b).
- Leer un punto en el espacio ($P = (x, y, z)$)
- Verificar si el punto ($P = (x, y, z)$) pertenece al paraboloide e informar al usuario el resultado de la verificación. Si el punto pertenece al paraboloide el programa debe emitir el mensaje `El punto pertenece al paraboloide`, en caso contrario debe emitir el mensaje `El punto no pertenece al paraboloide`.

Casos de prueba para verificar la solución

Use los siguiente casos para verificar si la solución está correcta.

a	b	x	y	z	Impresión
1	1	3	4	3	El punto no pertenece al paraboloide.
1	1	2	1	4	El punto no pertenece al paraboloide.
1	1	1	1	2	El punto pertenece al paraboloide.
2	3	16	9	73	El punto pertenece al paraboloide.

Variación al problema

Desarrolle el programa para un **hiperboloide**. Una de las formas canónicas del hiperboloide es:

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 - \left(\frac{z}{c}\right)^2 = 1$$

Use los siguientes casos de prueba para verificar si la solución está correcta. Debe leer los coeficientes a, b y c; números reales diferentes de cero.

a	b	c	x	y	z	Impresión
6	5	3	3	4	3	El punto no pertenece al hiperboloide.
5	1	2	19	9	4	El punto no pertenece al hiperboloide.
2	2	4	2	2	4	El punto pertenece al hiperboloide.
2	3	1	0	3	0	El punto pertenece al hiperboloide.

Variación al problema

Desarrolle el programa para un **elipsoide**. Una de las formas canónicas del elipsoide es:

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 + \left(\frac{z}{c}\right)^2 = 1$$

Use los siguientes casos de prueba para verificar si la solución está correcta. Debe leer los coeficientes a, b y c; números reales diferentes de cero.

a	**b**	**c**	**x**	**y**	**z**	**Impresión**
6	5	3	3	4	3	El punto no pertenece al elipsoide.
5	1	2	19	9	4	El punto no pertenece al elipsoide.
1	1	1	1	0	0	El punto pertenece al elipsoide.
0.5	0.5	0.5	0	0.5	0	El punto pertenece al elipsoide.

2.3.17. Ecuación bicuadrada (adaptado del laboratorio 3 2020-1)

La ecuación bicuadrada es una expresión de la siguiente forma:

$$ax^4 + bx^2 + c = 0$$

Donde los coeficientes (a, b y c) son números reales y además se debe cumplir necesariamente que a es diferente de cero. La ecuación bicuadrada posee 4 soluciones reales dependiendo del discriminante, pero para este problema asuma que siempre existirán las 4 soluciones reales. Use los casos de prueba de esta hoja para verifica su solución.

Las raíces serán:

$$x_1 = +\sqrt{\frac{-b+\sqrt{b^2-4ac}}{2a}} \quad x_2 = +\sqrt{\frac{-b-\sqrt{b^2-4ac}}{2a}}$$
$$x_3 = -\sqrt{\frac{-b+\sqrt{b^2-4ac}}{2a}} \quad x_4 = -\sqrt{\frac{-b-\sqrt{b^2-4ac}}{2a}}$$

El programa deberá realizar lo siguiente:

- Leer los coeficientes.
- Verificar si la ecuación es bicuadrada.
- En caso que la ecuación no sea bicuadrada debe emitir el siguiente mensaje `La ecuación no es bicuadrada`. En caso contrario debe calcular e imprimir cada una de las 4 raíces.
- *No debe repetir cálculos, debe usar variables intermedias siempre que lo necesite.*

.

Casos de prueba para verificar la solución

Use los siguiente casos para verificar si la solución está correcta.

a	**b**	**c**	x_1	x_2	x_3	x_4
1	-13	36	3	2	-3	-2
1	-10	9	3	1	-3	-1
1	-61	900	6	5	-6	-5

2.3.18. Ángulo entre vectores (adaptado del laboratorio 2 2021-1)

Dados dos vectores que parten del origen en un plano cartesiano, se pide que elabore un algoritmo expresado en pseudocódigo que determine el ángulo que se forma entre estos dos vectores.

El algoritmo deberá realizar lo siguiente:

- Leer un punto $P_1(x_1, y_1)$ que representará al vector 1.
- Leer un punto $P_2(x_2, y_2)$ que representará al vector 2.
- Calcular el producto punto entre los dos vectores.
- Calcular el módulo de cada vector.
- Usando el producto punto y el módulo de cada vector, deberá calcular el coseno del ángulo formado.
- Calcular el ángulo en grados sexagesimales y mostrarlo en pantalla. Recuerde que la función `acos` en PSeInt retornar el ángulo en radianes.
- Determinar e imprimir si el ángulo es agudo. Si el ángulo es agudo se deberá imprimir el siguiente mensaje `El ángulo es agudo`. En caso contrario se mostrará el mensaje `El ángulo es recto u obtuso`.

.

Recordar que:

Para transformar angulos de radianes a sexagesimales puede usar la siguiente relación:

$$360° = 2\pi \text{ radianes}$$

El producto punto se calcula de la siguiente manera:

$$\vec{u}(x_1, y_1) \cdot \vec{v}(x_2, y_2) = x_1 \times x_2 + y_1 \times y_2$$

El módulo de un vector se calcula de la siguiente manera:

$$|\vec{u}(x, y)| = \sqrt{x^2 + y^2}$$

El coseno entre los vectores $\vec{u}$ y $\vec{v}$ se calcula de la siguiente manera:

$$\cos(\theta) = \frac{\vec{u} \cdot \vec{v}}{|\vec{u}| \times |\vec{v}|}$$

A continuación siguen algunos ejemplos de ejecución del algoritmo:

Caso de prueba 1

```
*** Ejecución Iniciada. ***
Ingrese primer vector:
> 4
> -1
Ingrese segundo vector vector:
> 2
> 5
El ángulo entre los vectores es: 82.2348339816
El ángulo es agudo
*** Ejecución Finalizada. ***
```

Caso de prueba 2

*** Ejecución Iniciada. ***
Ingrese primer vector:
> 5
> 3
Ingrese segundo vector vector:
> 1
> 2
El ángulo entre los vectores es: 32.4711922908
El ángulo es agudo
*** Ejecución Finalizada. ***

Caso de prueba 3

*** Ejecución Iniciada. ***
Ingrese primer vector:
> -2
> -7
Ingrese segundo vector vector:
> -1
> 5
El ángulo entre los vectores es: 152.7446716251
El ángulo es recto u obtuso
*** Ejecución Finalizada. ***

2.3.19. Propiedades de logaritmos (adaptado del laboratorio 2 2021-1)

Los logaritmos poseen ciertas propiedades, entre ellas se encuentran las siguientes:

- $\log_{base}(a \times b) = \log_{base}(a) + \log_{base}(b)$
- $\log_{base}(\frac{a}{b}) = \log_{base}(a) - \log_{base}(b)$

Se pide que elabore un algoritmo expresado en pseudocódigo que permita verificar si ambas propiedades se cumplen.

El algoritmo deberá realizar lo siguiente:

- Leer la base.
- Leer los números a y b
- Calcular e imprimir en pantalla los valores de $\log_{base}(a \times b)$ y $\log_{base}(\frac{a}{b})$.
- Calcular por separado los valores de $\log_{base}(a)$ y $\log_{base}(b)$.
- Usando los valores calculados diseñe las expresiones lógicas que permitan determinar si se cumple cada una de las propiedades especificadas en este enunciado.
- Determinar, usando las expresiones lógicas calculadas, si se cumplen las propiedades. Si las propiedades se pueden verificar mediante el algoritmo, se debe imprimir el siguiente mensaje `Se cumplen las propiedades`. En caso contrario se mostrará el mensaje `No se cumplen las propiedades`.

.

Recordar que:

PSeInt solamente incluye la función `ln` que implementa el logaritmo natural. Si necesita obtener el logaritmo en otra base, debe usar el teorema de cambio de base. En particula puede usar la siguiente relación:

$$\log_{base}(x) = \frac{ln(x)}{ln(base)}$$

A continuación siguen algunos ejemplos de ejecución del algoritmo:

```
*** Ejecución Iniciada. ***
Ingrese la base:
> 2
Ingrese dos números:
> 8
> 4
logaritmo del producto = 5
logaritmo del cociente = 1
Se cumplen las propiedades
*** Ejecución Finalizada. ***

*** Ejecución Iniciada. ***
Ingrese la base:
> 4
Ingrese dos números:
> 6
> 7
logaritmo del producto = 2.6961587114
logaritmo del cociente = -0.1111962107
Se cumplen las propiedades
*** Ejecución Finalizada. ***

*** Ejecución Iniciada. ***
Ingrese la base:
> 3
Ingrese dos números:
> 27
> 81
logaritmo del producto = 7
logaritmo del cociente = -1
Se cumplen las propiedades
*** Ejecución Finalizada. ***
```

2.3.20. Trapezoide simétrico (adaptado del laboratorio 2 2021-1)

Dados 4 puntos en el plano cartesiano que representan los vértices de un trapezoide, se pide que elabore un algoritmo expresado en pseudocódigo que determine si el trapezoide es simétrico o no. Un trapezoide es simétrico si tiene los dos pares de lados consecutivos iguales.

El algoritmo deberá realizar lo siguiente:

- Leer los 4 puntos, $P_1(x_1, y_1)$, $P_2(x_2, y_2)$, $P_3(x_3, y_3)$ y $P_4(x_4, y_4)$, que representan a los vértices consecutivos de un trapezoide.
- Calcular los 4 lados del trapezoide considerando que se han ingresado los vértices de forma consecutiva.

- Determinar e imprimir si el trapezoide es simétrico o no. Si el trapezoide es simétrico se deberá imprimir el siguiente mensaje `Es un trapezoide simétrico`. En caso contrario se mostrará el mensaje `No es un trapezoide simétrico`. Para fines de esta pregunta, asuma que los lados consecutivos son i) el formado por los vértices 1, 2 y 3 y ii) el formado por los vértices 1, 4 y 3.

.

Recuerdar que:

Recuerde que la distancia euclidiana d se calcula de la siguiente manera:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

A continuación siguen algunos ejemplos de ejecución del algoritmo:

Caso de prueba 1

```
*** Ejecución Iniciada. ***
Ingrese punto 1:
> -2
> 0
Ingrese punto 2:
> 0
> 3
Ingrese punto 3:
> 2
> 0
Ingrese punto 4:
> 0
> -4
Es un trapezoide simétrico
*** Ejecución Finalizada. ***
```

Caso de prueba 2

```
*** Ejecución Iniciada. ***
Ingrese punto 1:
> -2
> -1
Ingrese punto 2:
> 2
> 3
Ingrese punto 3:
> 2
> 6
Ingrese punto 4:
> -1
> -7
No es un trapezoide simétrico
*** Ejecución Finalizada. ***
```

Caso de prueba 3

*** Ejecución Iniciada. ***
Ingrese punto 1:
> -2
> -3
Ingrese punto 2:
> 4
> 6
Ingrese punto 3:
> 1
> 5
Ingrese punto 4:
> -4
> -5
No es un trapezoide simétrico
*** Ejecución Finalizada. ***

2.3.21. Tipos de cuadriláteros (adaptado del laboratorio 2 2021-1)

Dados 4 puntos en el plano cartesiano que representan los vértices de un cuadrilátero, se pide que elabore un algoritmo expresado en pseudocódigo que determine si el cuadrilátero es un rombo. Un cuadrilátero es clasificado como rombo si tiene los 4 lados iguales pero dos de sus ángulos son menores que los otros dos.

El algoritmo deberá realizar lo siguiente:

- Leer los 4 puntos, $P_1(x_1, y_1)$, $P_2(x_2, y_2)$, $P_3(x_3, y_3)$ y $P_4(x_4, y_4)$, que representan a los vértices consecutivos de un cuadrilátero.
- Calcular los 4 lados del cuadrilátero considerando que se han ingresado los vértices de forma consecutiva.
- Determinar e imprimir si el cuadrilátero es un rombo o no. Si el cuadrilátero se puede clasificar como rombo, deberá imprimir el siguiente mensaje `Es un rombo`. En caso contrario se mostrará el mensaje `No es un rombo`. Para fines de esta pregunta, asuma que los lados consecutivos son i) el formado por los vértices 1, 2 y 3 y ii) el formado por los vértices 1, 4 y 3. Asuma además que el usuario nunca ingresará un cuadrado.

.

Recuerdar que:

Recuerde que la distancia euclidiana d se calcula de la siguiente manera:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

A continuación siguen algunos ejemplos de ejecución del algoritmo:

Caso de prueba 1

*** Ejecución Iniciada. ***
Ingrese punto 1:
> -3
> 2
Ingrese punto 2:
> 2
> 2
Ingrese punto 3:
> 5
> -2
Ingrese punto 4:
> 0
> -2
Es un rombo
*** Ejecución Finalizada. ***

Caso de prueba 2

*** Ejecución Iniciada. ***
Ingrese punto 1:
> 1
> 4
Ingrese punto 2:
> 3
> 7
Ingrese punto 3:
> 5
> 4
Ingrese punto 4:
> 3
> 1
Es un rombo
*** Ejecución Finalizada. ***

Caso de prueba 3

*** Ejecución Iniciada. ***
Ingrese punto 1:
> -4
> 2
Ingrese punto 2:
> 1
> 4
Ingrese punto 3:
> -1
> -1
Ingrese punto 4:
> -5
> 1
No es un rombo
*** Ejecución Finalizada. ***

2.3.22. Área del Pentágono (adaptado del laboratorio 2 2021-1)

Se desea comparar el área de dos figuras geométricas, un pentágono regular y un círculo. El objetivo es deteminar cual de ellas es mayor. Se pide que elabore un algoritmo expresado en pseudocódigo que dado dos vértices consecutivos de un pentágono regular y el radio de un círculo, determine la figura que posee mayor área

El algoritmo deberá realizar lo siguiente:

- Leer los 2 vértices, $P_1(x_1, y_1)$ y $P_2(x_2, y_2)$ que representan a dos vértices consecutivos de un pentágono regular.
- Leer el radio del círculo.
- Calcular el lado del pentágono y calcular su área. El área de un pentágono regular es aproximadamente $1.72 \times lado^2$.
- Calcular el área del círculo.
- Determinar e imprimir que figura posee mayor área. Si es el pentágono regular se deberá imprimir el siguiente mensaje `El área del pentágono es mayor`. En caso contrario se mostrará el mensaje `El área del círculo es mayor o igual`. En ambos casos deberá imprimirse el área calculada.

.

A continuación siguen algunos ejemplos de ejecución del algoritmo:

Caso de prueba 1

*** Ejecución Iniciada. ***
Ingrese vértice 1:
> 1
> 2
Ingrese vértice 2:
> 4
> 5
Ingrese radio:
> 6
El área del círculo es mayor o igual 113.0973355292
*** Ejecución Finalizada. ***

Caso de prueba 2

*** Ejecución Iniciada. ***
Ingrese vértice 1:
> 1
> 2
Ingrese vértice 2:
> 4
> 5
Ingrese radio:
> 3
El área del pentágono es mayor 30.96
*** Ejecución Finalizada. ***

Caso de prueba 3

*** Ejecución Iniciada. ***
Ingrese vértice 1:
> 1
> 2
Ingrese vértice 2:
> 3
> 4
Ingrese radio:
> 5
El área del círculo es mayor o igual 78.5398163397
*** Ejecución Finalizada. ***

2.3.23. Movimiento parabólico (adaptado del laboratorio 2 2021-2)

El movimiento parabólico corresponde al movimiento de un objeto que sigue una trayectoria parabólica. El medio en el que se mueve el objeto no ofrece resistencia y está sujeto a la gravedad.

Como se muestra en la figura 2.2; si se conoce la velocidad inicial ($v_{inicial}$), el ángulo de lanzamiento ($ngulo$) y la gravedad ($\approx 9.81\ m/s^2$); y utilizando el movimiento rectilíneo uniforme en el eje horizontal y el movimiento rectilíneo uniformemente acelerado en el eje (considerando la gravedad como la aceleración); se tienen las formulas descritas en la tabla "Fórmulas del movimiento parabólico".

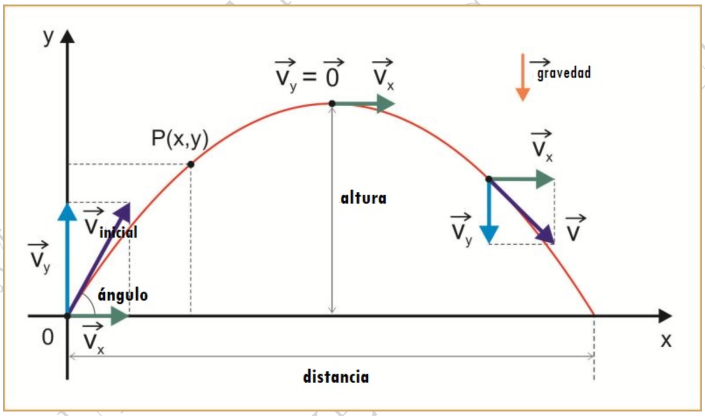

Figura 2.2: Trayectoria de un proyectil. Imagen basada en la URL http://osfundamentosdafisica.blogspot.pe/2011/01/resolucao-de-preparando-se-para-as_22.html.

Fórmulas del movimiento parabólico

$$distancia = \frac{(v_{inicial})^2 \times sen(2 \times angulo)}{gravedad}$$

$$altura = \frac{(v_{inicial})^2 \times (sen(angulo))^2}{2 \times gravedad}$$

$$tiempo = \frac{2 * v_{inicial} \times sen(angulo)}{gravedad}$$

Despejando el ángulo en la fórmula del tiempo:

$$angulo = asen(\frac{tiempo \times gravedad}{2 \times v_{inicial}})$$

La distancia y la altura se miden en metros (m), la velocidad en metros/segundo (m/s), el ángulo en radianes (rad), la gravedad en metros/segundo^2 (m/s^2) y el tiempo en segundos (s).

Se pide que desarrolle un **pseudocódigo** que analice dos escenarios de un movimiento parabólico para el lanzamiento de una pelota.

- Para el primer escenario, debe:
  - solicitar la velocidad inicial en metros/segundos (m/s) y el ángulo de tiro en grados sexagesimales
  - validar si la velocidad es positiva y si el ángulo es positivo y menor de 360°.
  - luego de la validación de los datos, calcule la altura en metros, la distancia en kilómetros y el tiempo en segundos.
- Para el segundo escenario, debe:
  - solicitar la velocidad inicial en kilómetros/hora (km/h) y el tiempo en minutos (min)
  - validar que ambos valores sean positivos.
  - luego de la validación de los datos, debe calcular el ángulo en grados sexagesimales, la altura en metros y la distancia en kilómetros.
- Finalmente, si os datos son válidos en los dos escenarios, debe identificar en cuál de ellos la pelota alcanza una mayor altura.

Recuerde:

- para realizar los cálculos correctos debe usar las mismas unidades, por lo cual debe realizar las conversiones necesarias.

En PseInt la función sen cálcula el seno de un ángulo dado en radianes y la función asen calcula el arcoseno de un número devolviendo el ángulo en radianes.

Conversión de unidades

360°=2π radianes
1 minuto = 60 segundos
1 kilómetro/hora = 5/18 metros/segundo
π = 3.141592

Use los siguientes casos de prueba para verificar si su solución está correcta.

Caso de prueba 1

```
*** Ejecución Iniciada. ***
Datos escenario 1:
Ingrese la velocidad inicial de la pelota (m/s):
> 26
Ingrese el ángulo de tiro de la pelota (grados sexagesimales):
> 80
Resultados Escenario 1
Altura: 33.4157036608
Distancia: 0.0235683605 km
Tiempo: 5.220183808 s
Datos Escenario 2
Ingrese la velocidad inicial de la pelota (km/h):
> 200
Ingrese el tiempo que la pelota se encuentra en el aire (minutos):
> 0.11
Resultados Escenario 2
Ángulo: 35.6416582249 grados sexagesimales
Altura: 53.41545 m
Distancia: 0.2979816767 km
En el escenario 2 la pelota alcanza una altura mayor
la diferencia entre las alturas es: 19.9997463392 m
*** Ejecución Finalizada. ***
```

Caso de prueba 2

```
*** Ejecución Iniciada. ***
Datos escenario 1:
Ingrese la velocidad inicial de la pelota (m/s):
> 12
Ingrese el ángulo de tiro de la pelota (grados sexagesimales):
> -5
Para el escenario 1 los datos no son válidos
Datos Escenario 2
Ingrese la velocidad inicial de la pelota (km/h):
> 180
Ingrese el tiempo que la pelota se encuentra en el aire (minutos):
> 0.15
Resultados Escenario 2
Ángulo: 61.9941921721 grados sexagesimales
Altura: 99.32625 m
Distancia: 0.2113024774 km
*** Ejecución Finalizada. ***
```

Caso de prueba 3

*** Ejecución Iniciada. ***
Datos escenario 1:
Ingrese la velocidad inicial de la pelota (m/s):
> 20
Ingrese el ángulo de tiro de la pelota (grados sexagesimales):
> 75
Resultados Escenario 1
Altura: 19.0216656859
Distancia: 0.0203873598 km
Tiempo: 3.9385354793 s
Datos Escenario 2
Ingrese la velocidad inicial de la pelota (km/h):
> -4
Ingrese el tiempo que la pelota se encuentra en el aire (minutos):
> 10
Para el escenario 2 los datos no son válidos
*** Ejecución Finalizada. ***

Caso de prueba 4

*** Ejecución Iniciada. ***
Datos escenario 1:
Ingrese la velocidad inicial de la pelota (m/s):
> 40
Ingrese el ángulo de tiro de la pelota (grados sexagesimales):
> 89.9
Resultados Escenario 1
Altura: 81.5491909341
Distancia: 0.0005693213 km
Tiempo: 8.1549315141 s
Datos Escenario 2
Ingrese la velocidad inicial de la pelota (km/h):
> 180
Ingrese el tiempo que la pelota se encuentra en el aire (minutos):
> 0.09
Resultados Escenario 2.
Ángulo: 31.9878893593 grados sexagesimales
Altura: 35.75745 m
Distancia: 0.2290032234 km
En el escenario 1 la pelota alcanza una altura mayor o igual
la diferencia entre las alturas es: 45.7917409341 m
*** Ejecución Finalizada. ***

2.3.24. Elipse (adaptado del laboratorio 2 2021-2)

En la Figura 2.3 se muestra una elipse en el plano cartesiano con semiejes semieje_a y semieje_b y centro en el punto (centrox,centroy).

En la figura 2.4 se muestra una elipse donde el punto P1(x1,y1) se encuentra dentro de la elipse, el punto P2(x2,y2) se encuentra en la elipse y el punto P3(x3,y3) se encuentra fuera de la elipse. Un punto pertenece a la elipse cuando este se evalúa en la ecuación ordinaria de la elipse siendo el resultado 1, el punto se encuentra dentro de la elipse si el resultado es menor a 1 y el punto se encuentra fuera de la elipse si el resultado es mayor a 1.

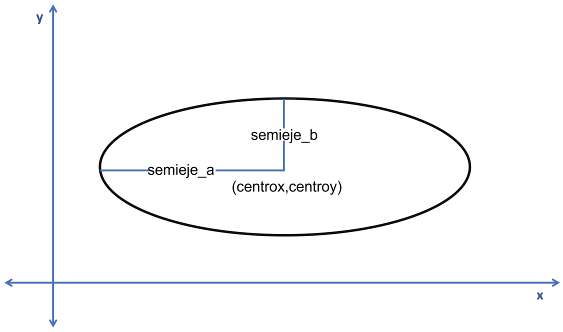

Figura 2.3: Elipse

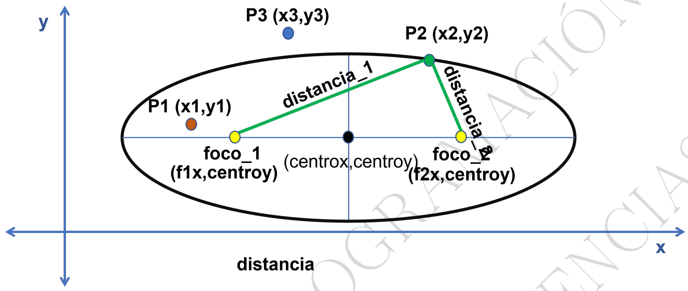

Figura 2.4: Focos de la elipse

En la figura 2.4 también se muestran los focos de la elipse foco_1 y foco_2, los cuales son puntos fijos que generan la elipse. La suma de las distancias de cualquier punto de la elipse a los focos es igual a 2 veces el semieje_a: distancia_1 + distancia_2 = 2 × semieje_a, esta afirmación se conoce como la propiedad de la elipse. Para calcular las coordenadas de los focos tenga en cuenta que la ordenada es la misma y las abcisas se obtienen con las fórmulas descritas en la tabla ”Ecuaciones de la elipse y fórmulas para calcular las abscisas de los focos”

Ecuaciones de la elipse y fórmulas para calcular las abscisas de los focos

Ecuación ordinaria de la elipse:

$$\frac{(x - centrox)^2}{semieje\_a^2} + \frac{(y - centroy)^2}{semieje\_b^2} = 1$$

Ecuación paramétrica de la elipse en el eje x:

$$x = centrox + semieje\_a \times cos(angulo)$$

Despejando angulo en la ecuación anterior:

$$angulo = acos(\frac{x - centrox}{semieje\_a})$$

Ecuación paramétrica de la elipse en el eje y:

$$y = centroy + semieje\_b \times sen(angulo)$$

Distancia del centro de la elipse (centrox) a los focos: (f1x y f2x):

$$distancia\_centro\_foco = \sqrt{semieje\_a^2 - semieje\_b^2}$$

Fórmula para calcular las abscisa del foco_1:

$$f1x = centrox - distancia\_centro\_foco$$

Fórmula para calcular las abscisa del foco_1:

$$f2x = centrox + distancia\_centro\_foco$$

Propiedad de la elipse:

$$distancia\_1 + distancia\_2 = 2 \times semieje\_a$$

Fórmulas para calcular el área y el perímetro

Fórmula para calcular el área

$$area = \pi \times semieje\_a \times semieje\_b$$

Fórmula 1, para calcular el perímetro:

$$perimetro = 2 \times \pi \times \sqrt{\frac{semieje\_a^2 + semieje\_b^2}{2}}$$

Fórmula 2, para calcular el perímetro:

$$perimetro \approx \pi \times (semieje\_a + semieje\_b) \times (1 + \frac{3 \times H}{10 + \sqrt{4 - 3 \times H}} + (\frac{4}{\pi} - \frac{14}{11}) \times H^{12})$$

Donde H, se calcula con:

$$H = (\frac{semieje\_a - semieje\_b}{semieje\_a + semieje\_b})^2$$

Se pide que desarrolle un **pseudocódigo** que:

- solicite el punto del plano cartesiano del centro de la elipse y los semiejes respectivos
- valide que los semiejes son positivos y que el semieje_a es mayor que el semieje_b
- si los datos son válidos calcule el área y el perímetro de la misma. Para el perímetro debe usar las dos fórmulas descritas anteriormente y mostrar la diferencia entre los resultados.
- si los datos no son válidos muestre el mensaje "Los semiejes ingresados deben ser positivos y el semieje_a mayor que el semieje_b"
- si los datos fueron válidos, luego de realizar los cálculos de área y perímetro, solicite un punto en el plano cartesiano y determine si el punto se encuentra dentro de la elipse, pertenece a la elipse o se encuentra fuera de la elipse.
- finalmente, si el punto pertenece a la elipse calcule el valor del ángulo de las ecuaciones paramétricas de la elipse (en grados sexagesimales), calcule los puntos de los focos y muestre si se cumple la propiedad de la elipse, evaluando la fórmula respectiva desde el punto ingresado. Recuerde que $360° = 2\pi$ radianes.

En PseInt la función acos calcula el arcoseno de un número devolviendo el ángulo en radianes y la función abs calcula el valor absoluto de un número.

Para calcular el área y el perímetro de la elipse use las fórmulas descritas en la tabla ”Fórmulas para calcular el área y el perímetro”

Recuerde que el resultado de la comparación de números reales a través de la igualdad no es preciso, por lo cual debe usar el valor absoluto de la diferencia de los números que se desean comparar. Si esta diferencia es cercana a cero, se puede asumir que son iguales, para realizar esta comparación use el valor 0.0001.

Use los siguientes casos de prueba para verificar si su solución está correcta.

Caso de prueba 1

```
*** Ejecución Iniciada. ***
Ingrese el centro de la elipse:
> 3
> -1
Ingrese los semiejes:
> 5
> 4
Cálculos de la elipse:
El área de la elipse es 62.8318530718
El perímetro de la elipse según la fórmula 1 es 28.4483314254
El perímetro de la elipse según la fórmula 2 es 28.361667889
La diferencia entre los perímetros es 0.0866635364
Ingrese un punto para verificar su posición
> 5.5
> 2.46410118
El punto ingresado es parte de la elipse
El ángulo de las ecuaciones paramétricas de la elipse es 60 grados sexagesimales
Coordenadas de los focos: (0,-1) y (6,-1)
Se cumple la propiedad de la elipse
*** Ejecución Finalizada. ***
```

Caso de prueba 2

```
*** Ejecución Iniciada. ***
Ingrese el centro de la elipse:
> 2.5
> 1
Ingrese los semiejes:
> 6
> 3.5
Cálculos de la elipse:
El área de la elipse es 65.9734457254
El perímetro de la elipse según la fórmula 1 es 30.861251185
El perímetro de la elipse según la fórmula 2 es 30.3641157796
La diferencia entre los perímetros es 0.4971354054
Ingrese un punto para verificar su posición
> 4
> 4.3888
El punto ingresado es parte de la elipse
El ángulo de las ecuaciones paramétricas de la elipse es 75.5224878141 grados sexagesimales
Coordenadas de los focos: (-2.3733971724,1) y (7.3733971724,1)
Se cumple la propiedad de la elipse
*** Ejecución Finalizada. ***
```

Caso de prueba 3

```
*** Ejecución Iniciada. ***
Ingrese el centro de la elipse:
> 2
> 2
Ingrese los semiejes:
> 6.5
> 2.5
Cálculos de la elipse:
El área de la elipse es 51.0508806208
El perímetro de la elipse según la fórmula 1 es 30.9410993164
El perímetro de la elipse según la fórmula 2 es 29.6887600509
La diferencia entre los perímetros es 1.2523392655
Ingrese un punto para verificar su posición
> 1
> 1
El punto ingresado está dentro de la elipse
*** Ejecución Finalizada. ***
```

Caso de prueba 4

*** Ejecución Iniciada. ***
Ingrese el centro de la elipse:
> -3.4
> -2.1
Ingrese los semiejes:
> 5.5
> 2.2
Cálculos de la elipse:
El área de la elipse es 38.0132711084
El perímetro de la elipse según la fórmula 1 es 26.3182225249
El perímetro de la elipse según la fórmula 2 es 25.3144240982
La diferencia entre los perímetros es 1.0037984267
Ingrese un punto para verificar su posición
> 10
> 10
El punto ingresado está fuera de la elipse
*** Ejecución Finalizada. ***

Caso de prueba 5

*** Ejecución Iniciada. ***
Ingrese el centro de la elipse:
> -2.5
> -3.4
Ingrese los semiejes:
> 4
> 7.5
Los semiejes ingresados deben ser positivos y el semieje_a mayor que el semieje_b.
*** Ejecución Finalizada. ***

2.3.25. Movimiento armónico simple (adaptado del laboratorio 2 2021-2)

El movimiento armónico simple (M.A.S), como se muestra en la figura 2.5, corresponde al movimiento de un móvil oscilatorio y periódico, la aceleración señala hacia la posición de equilibrio y es directamente proporcional a la distancia del objeto a la posición de equilibrio.

Las fórmulas asociadas a este movimiento se describen en la tabla "Fórmulas del movimiento armónico simple".

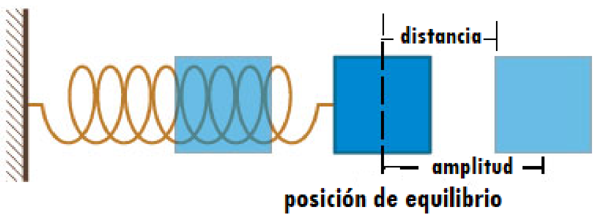

Figura 2.5: Movimiento armónico simple. Imagen basada en la URL https://matemovil.com/movimiento-armonico-simple-mas-ejercicios-resueltos/.

Fórmulas del movimiento armónico simple

$$distancia = amplitud * sen(frecuencia * tiempo + fase)$$

$$velocidad = amplitud * frecuencia * cos(frecuencia * tiempo + fase)$$

$$velocidad = frecuencia * \sqrt{amplitud^2 - distancia^2}$$

$$aceleracion = -amplitud * frecuencia^2 * cos(frecuencia * tiempo + fase)$$

Despejando la fase en la fórmula de la velocidad:

$$fase = acos(velocidad/(amplitud * frecuencia)) - frecuencia * tiempo$$

Donde: frecuencia, corresponde a la frecuencia cícilica y fase es la fase inicial del M.A.S . La distancia y la amplitud se miden en metros (m), la frecuencia en radianes/segundo (rad/s), el tiempo en segundos (s), la fase inicial en radianes (rad), la velocidad en metros/segundo (m/s) y la aceleración en metros/segundo^2 (m/s^2).

Se pide que desarrolle un **pseudocódigo** que analice dos escenarios de un movimiento armónico simple para un móvil.

- Para el primer escenario, debe:
  - solicitar la magnitud en metros (m), la frecuencia cíclica en radianes/segundo (rad/s), la fase inicial en grados sexagesimales y el tiempo en minutos (min).
  - validar si todos los datos son positivos y si la fase inicial es menor de 360°.
  - luego de la validación de los datos, calcule la distancia en metros, la velocidad en metros/segundo y la aceleración en metros/segundo^2.
- Para el segundo escenario, debe:
  - solicitar la velocidad inicial en metros/hora (m/h), la amplitud en metros(m), la distancia en centímetros (cm) y el tiempo de vuelo en segundos (s)
  - validar que ambos valores sean positivos y que la amplitud sea menor de 100 m.
  - luego de la validación de los datos, debe calcular la frecuencia cíclica en radianes/segundo, la fase inicial en radianes y la aceleración en metros/segundo^2..
- Finalmente, si los datos ingresados para los dos escenarios fueron válidos, debe identificar en cuál de ellos la aceleración del móvil es mayor.

Recuerde:

- para realizar los cálculos correctos debe usar las mismas unidades, por lo cual debe realizar las conversiones necesarias.

En PseInt la función sen cálcula el seno de un ángulo dado en radianes y la función acos calcula el arcocoseno de un número devolviendo el ángulo en radianes.

Conversión de unidades

360°=2π radianes
1 minuto = 60 segundos
1 hora = 3600 segundos
1 metro = 100 centímetros
π = 3.141592

Use los siguientes casos de prueba para verificar si su solución está correcta.

Caso de prueba 1

*** Ejecución Iniciada. ***
Datos Escenario 1:
Ingrese la amplitud (m):
> 0.4
Ingrese la frecuencia cíclica (rad/s):
> 3.141592
Ingrese la fase inicial (grados sexagesimales):
> 45
Ingrese el tiempo (min):
> 0.5
Resultados Escenario 1
Distancia 0.2828371665 m
Velocidad 0.8885938255 m/s
Aceleración -2.7915445183 m/s2
Datos Escenario 2
Ingrese la velocidad (m/h):
> 1500
Ingrese la amplitud (m):
> 0.5
Ingrese la distancia (cm):
> 25
Ingrese el tiempo (s):
> 32
Resultados Escenario 2
Frecuencia cíclica 0.9622504486 rad/s
Fase inicial -30.2684155812 rad
Aceleración -0.2314814815 m/s2
En el escenario 2 la aceleración es mayor o igual que el escenario 1, la diferencia entre las aceleraciones es: 2.5600630368 m/s2
*** Ejecución Finalizada. ***

Caso de prueba 2

```
*** Ejecución Iniciada. ***
Datos Escenario 1:
Ingrese la amplitud (m):
> 13
Ingrese la frecuencia cíclica (rad/s):
> -45
Ingrese la fase inicial (grados sexagesimales):
> -89.6
Ingrese el tiempo (min):
> -3
Para el escenario 1 los datos no son válidos
Datos Escenario 2
Ingrese la velocidad (m/h):
> 12
Ingrese la amplitud (m):
> 4.5
Ingrese la distancia (cm):
> 120
Ingrese el tiempo (s):
> 34
Resultados Escenario 2
Frecuencia cíclica 0.0007685716 rad/s
Fase inicial 0.2438013605 rad
Aceleración -0.0000007088 m/s2
*** Ejecución Finalizada. ***
```

Caso de prueba 3

```
*** Ejecución Iniciada. ***
Datos Escenario 1:
Ingrese la amplitud (m):
> 2.5
Ingrese la frecuencia cíclica (rad/s):
> 1.8
Ingrese la fase inicial (grados sexagesimales):
> 5
Ingrese el tiempo (min):
> 3.9
Resultados Escenario 1
Distancia 0.7718092232 m
Velocidad 4.2801829511 m/s
Aceleración -2.5006618831 m/s2
Datos Escenario 2
Ingrese la velocidad (m/h):
> -3.7
Ingrese la amplitud (m):
> 5.6
Ingrese la distancia (cm):
> 10
Ingrese el tiempo (s):
> -2
Para el escenario 2 los datos no son válidos
*** Ejecución Finalizada. ***
```

Caso de prueba 4

```
*** Ejecución Iniciada. ***
Datos Escenario 1:
Ingrese la amplitud (m):
> 1.1
Ingrese la frecuencia cíclica (rad/s):
> 0.7532
Ingrese la fase inicial (grados sexagesimales):
> 0.59
Ingrese el tiempo (min):
> 0.76
Resultados Escenario 1
Distancia 0.2199394568 m
Velocidad 0.8117898036 m/s
Aceleración -0.124773906 m/s2
Datos Escenario 2
Ingrese la velocidad (m/h):
> 4240
Ingrese la amplitud (m):
> 0.8
Ingrese la distancia (cm):
> 27.9
Ingrese el tiempo (s):
> 36.5
Resultados Escenario 2
Frecuencia cíclica 1.5708465836 rad/s
Fase inicial -56.9796632672 rad
Aceleración -0.688448958 m/s2
En el escenario 2 la aceleración es menor que el escenario 1
la diferencia entre las aceleraciones es: 0.563675052 m/s2
*** Ejecución Finalizada. ***
```

2.3.26. Semejanza de triángulos (adaptado del laboratorio 2 2021-2)

Los triángulos que se muestran en la Figura 2.6 son semejantes si se cumplen algunas de las afirmaciones de la tabla "Semejanza de triángulos".

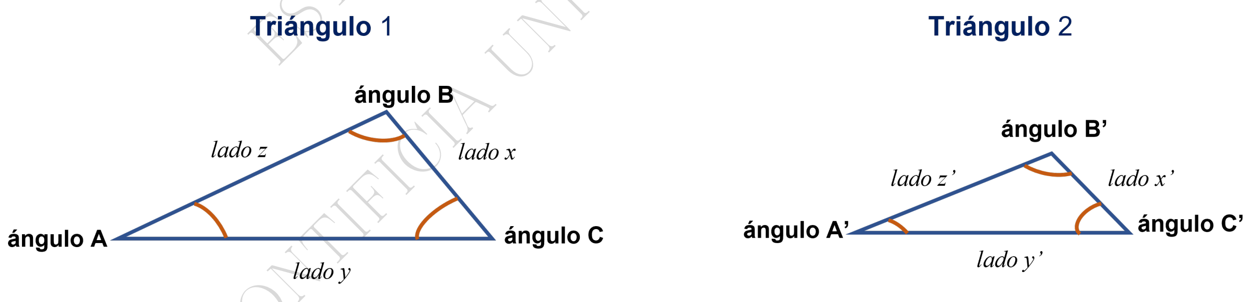

Figura 2.6: Semejanza de triángulos

Semejanza de triángulos

Tienen dos ángulos iguales, como:

$$anguloA = anguloA'$$

$$anguloB = anguloB'$$

Tienen sus tres lados proporcionales

$$\frac{ladox}{ladox'} = \frac{ladoy}{ladoy'} = \frac{ladoz}{ladoz'}$$

Tienen 2 lados proporcionales y el ángulo que forman estos lados es igual

$$\frac{ladox}{ladox'} = \frac{ladoy}{ladoy'}$$

$$anguloC = anguloC'$$

Las razones trigonométricas en un triángulo rectángulo se muestran en la tabla 2.3.26 .

Razones trigonométricas

$$sen(angulo) = \frac{cateto\_opuesto}{hipotenusa}$$

Despejando cateto_opuesto en la ecuación anterior:

$$cateto\_opuesto = sen(angulo) \times hipotenusa$$

$$cos(angulo) = \frac{cateto\_contiguo}{hipotenusa}$$

Despejando cateto_contiguo en la ecuación anterior:

$$cateto\_contiguo = cos(angulo) \times hipotenusa$$

En la figura 2.7, que describe el escenario 1, se muestran dos postes cuya ubicación y la línea de sus respectivas sombras forman 2 triángulos que cumplen con la semejanza de triángulos.

En la figura 2.8, se complementa el escenario 1 con un edificio que forma un triángulo rectángulo.

Se pide que desarrolle un **pseudocódigo** que analice los escenarios descritos en las figuras 2.7 y 2.8, usando la semejanza de triángulo y las razones trigonométricas.

- Para el primer escenario descrito en la figura 2.7, debe:
  - solicitar la altura del poste 1 en centímetros (cm), el seno del ángulo poste (formado entre el piso y la línea de la sombra de los edificios), la distancia entre los postes en metros (m) y la distancia entre el segundo poste y el punto del ángulo en centímetros (cm).
  - validar si todos los datos son positivos y si el seno del ángulo es menor que 1
  - luego de la validación de los datos, calcule la altura del segundo poste en metros, la distancia del primer poste al punto del ángulo en metros, el ángulo que se forma entre el piso y la línea de sombra de los edificios en grados sexagesimales, el ángulo que se forma entre los edificios y la línea de sombra de los edificios en grados sexagesimales y la longitud de la línea de la sombra en metros.
- Para la parte adicional al primer escenario, que se muestra en el segundo escenario descrito en la figura 2.7, debe:

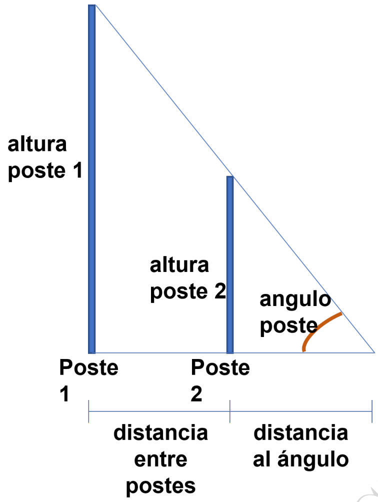

Figura 2.7: Escenario 1

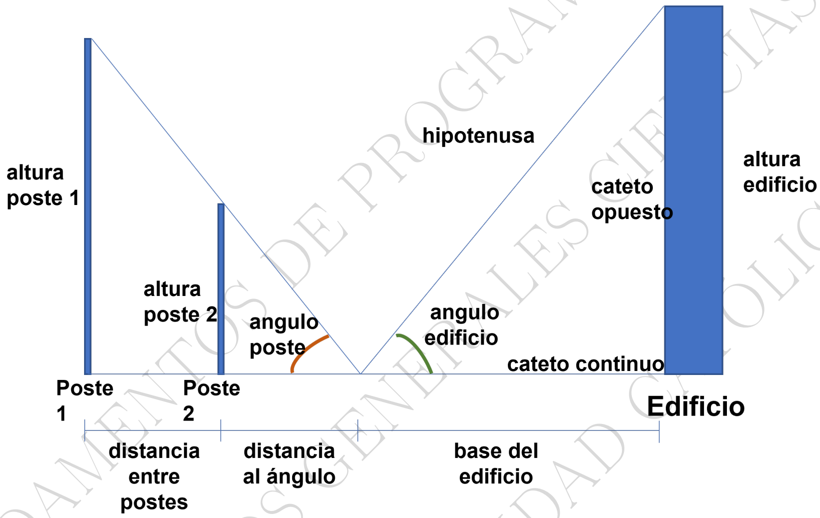

Figura 2.8: Escenario 2

- solicitar la hipotenusa del triángulo en decímetros y el ángulo eficio en grados sexagesimales.
- validar que ambos valores sean positivos y que el ángulo sea menor a 90 grados sexagesimales.
- luego de la validación de los datos, debe calcular la altura y base del edificio en metros.

- Finalmente, si los datos ingresados para los dos escenarios fueron válidos, debe identificar si la altura del edifcio es mayor o igual que las alturas de los postes o si es menor que ellos.

Recuerde:

- si tiene el seno de un ángulo puede usar el arcoseno para hallar el ángulo en radianes.
- el Teorema de Pitágoras: "el cuadrado de la hipotenusa es igual a la suma de los cuadrados de los catetos".
- para realizar los cálculos correctos debe usar las mismas unidades, por lo cual debe realizar las conversiones necesarias.

En PseInt la función sen cálcula el seno de un ángulo dado en radianes, la función cos calcula el coseno de un ángulo dado en radianes y la función asen calcula el arcoseno de un número devolviendo el ángulo en radianes.

Conversión de unidades

$360° = 2\pi$ radianes
1 metro $=$ 100 centímetros
1 metro $=$ 10 decímetros
$\pi = 3.141592$

Use los siguientes casos de prueba para verificar si su solución está correcta.

Caso de prueba 1

*** Ejecución Iniciada. ***
Datos del escenario 1
Ingrese la altura del poste 1 (en centímetros):
$>$ 1500
Ingrese el seno del ángulo entre el piso y la sombra de los edificios:
$>$ 0.5
Ingrese la distancia entre los postes (en metros):
$>$ 100
Ingrese la distancia entre el segundo poste y el punto del ángulo (en centímetros):
$>$ 50000
Resultados del escenario 1
La altura del segundo poste es 12.5 m
La distancia del primer poste al punto del ángulo es 600 m
El ángulo que se forma entre el piso y la línea de sombra de los edificios es 30 grados sexagesimales
El ángulo que se forma entre los edificios y la línea de sombra de los edificios es 60 grados sexagesimales
La longitud de la línea de la sombra es 600.1874707123 m
Datos del escenario 2
Ingrese la hipotenusa del triángulo (en decímetros):
$>$ 4000
Ingrese el ángulo inferior del triángulo (en grados sexagesimales):
$>$40
Resultado del escenario 2
La altura del edificio es 257.1150438746 m
La base del edificio es 306.4177772476 m
El edificio es más alto o de la misma altura que los postes
*** Ejecución Finalizada. ***

Caso de prueba 2

*** Ejecución Iniciada. ***
Datos del escenario 1
Ingrese la altura del poste 1 (en centímetros):
> 1000
Ingrese el seno del ángulo entre el piso y la sombra de los edificios:
> 2.34
Ingrese la distancia entre los postes (en metros):
> 50
Ingrese la distancia entre el segundo poste y el punto del ángulo (en centímetros):
> 1000
Los datos ingresados del escenario 1 no son válidos
Datos del escenario 2
Ingrese la hipotenusa del triángulo (en decímetros):
> 100
Ingrese el ángulo inferior del triángulo (en grados sexagesimales):
> 56.5
Resultado del escenario 2
La altura del edificio es 8.3388582207 m
La base del edificio es 5.5193698531 m
*** Ejecución Finalizada. ***

Caso de prueba 3

*** Ejecución Iniciada. ***
Datos del escenario 1
Ingrese la altura del poste 1 (en centímetros):
> 10000
Ingrese el seno del ángulo entre el piso y la sombra de los edificios:
> 0.23
Ingrese la distancia entre los postes (en metros):
> 35.6
Ingrese la distancia entre el segundo poste y el punto del ángulo (en centímetros):
> 20000
Resultados del escenario 1
La altura del segundo poste es 84.8896434635 m
La distancia del primer poste al punto del ángulo es 235.6 m
El ángulo que se forma entre el piso y la línea de sombra de los edificios es 13.2970717472 grados sexagesimales
El ángulo que se forma entre los edificios y la línea de sombra de los edificios es 76.7029282528 grados sexagesimales
La longitud de la línea de la sombra es 255.9440563873 m
Datos del escenario 2
Ingrese la hipotenusa del triángulo (en decímetros):
> 345
Ingrese el ángulo inferior del triángulo (en grados sexagesimales):
> -45
Los datos ingresados del escenario 2 no son válidos
*** Ejecución Finalizada. ***

Caso de prueba 4

*** Ejecución Iniciada. ***
Datos del escenario 1
Ingrese la altura del poste 1 (en centímetros):
> 1000
Ingrese el seno del ángulo entre el piso y la sombra de los edificios:
> 0.3
Ingrese la distancia entre los postes (en metros):
> 1
Ingrese la distancia entre el segundo poste y el punto del ángulo (en centímetros):
> 400
Resultados del escenario 1
La altura del segundo poste es 8 m
La distancia del primer poste al punto del ángulo es 5 m
El ángulo que se forma entre el piso y la línea de sombra de los edificios es 17.4576031237 grados sexagesimales
El ángulo que se forma entre los edificios y la línea de sombra de los edificios es 72.5423968763 grados sexagesimales
La longitud de la línea de la sombra es 11.1803398875 m
Datos del escenario 2
Ingrese la hipotenusa del triángulo (en decímetros):
> 145.5
Ingrese el ángulo inferior del triángulo (en grados sexagesimales):
> 32.3
Resultado del escenario 2
La altura del edificio es 7.7748266836 m
La base del edificio es 12.2985596734 m
El edificio tiene menos altura que los postes
*** Ejecución Finalizada. ***

2.3.27. Las capas de la esfera (adaptado del laboratorio 5 2021-1)

Se tiene una esfera que presenta 4 capas de distintos colores: rojo (R), azul (A), verde (V) y amarillo (M). El centro de dicha esfera se encuentra en el origen de coordenadas (0.00, 0.00, 0.00). (ver figura 2.9)

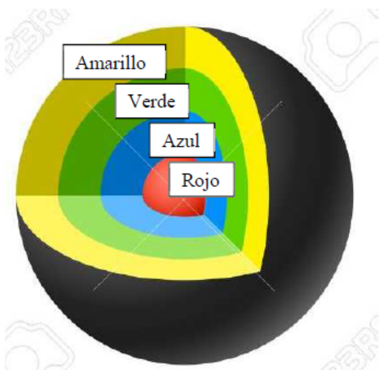

Figura 2.9: Capas

Se pide que elabore un programa en C que permita al usuario ingresar las coordenadas de un punto en el espacio y calcule la distancia del punto al origen de coordenadas e identificar el color de la capa donde está situado dicho punto.

Para hallar la distancia entre dos puntos en el espacio considerar la siguiente fórmula:

$$distancia = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Para identificar el color de la capa donde está situado un punto de coordenadas, considerar la tabla que se muestra en la figura 2.10)

Color de la capa	Rango de la distancia (cm)
Rojo	0 < distancia <= 5
Azul	5 < distancia <= 15
Verde	15 < distancia <= 30
Amarillo	30 < distancia

Figura 2.10: Rango de distancias

Caso de prueba 1

Ingrese las coordenadas del punto
Coordenada x: 1.00
Coordenada y: 2.00
Coordenada z: 3.00
La distancia del punto al origen es: 3.74
El punto cae en el área de color –R

Caso de prueba 2

Ingrese las coordenadas del punto
Coordenada x: 5.00
Coordenada y: 2.00
Coordenada z: 1.00
La distancia del punto al origen es: 5.48
El punto cae en el área de color -A

Caso de prueba 3

Ingrese las coordenadas del punto
Coordenada x: 5.00
Coordenada y: 5.00
Coordenada z: 22.00
La distancia del punto al origen es: 23.11
El punto cae en el área de color –V

2.3.28. Circunferencias y rectas (adaptado del laboratorio 5 2021-2)

Se le pide implementar un programa en lenguaje C que permita al usuario ingresar la letra “A” o “B” (también podría ingresar la letra en minúscula) para elegir entre: determinar si dos circunferencias se cruzan o determinar la relación entre dos rectas.

Si ingresa la opción "A" o "a" debe determinar si dos circunferencias de la forma $x^2 + y^2 + Cx + Dy + E = 0$ se cruzan; para ello, debe realizar lo siguiente:

- Solicitar al usuario que ingrese los coeficientes C, D y E de dos circunferencias de la forma $x^2 + y^2 + Cx + Dy + E = 0$.
- Deberá mostrar si las circunferencias se cruzan o no.

Si ingresa la opción "B" o "b" debe determinar si las rectas ingresadas son coincidentes, paralelas o secantes; para ello debe realizar lo siguiente:

- Solicitar al usuario que ingrese los coeficientes A, B y C de dos rectas de la forma $Ax + By + C = 0$.
- Deberá mostrar si las rectas son coincidentes, paralelas o secantes.

El programa debe mostrar mensajes específicos ante las siguientes situaciones:

- Al ingresar la opción debe verificar que sea "A" o "B" (también podría ingresar la letra en minúscula). En caso no se cumpla, se deberá emitir el siguiente mensaje "Error en la opción ingresada." y el programa debe terminar.
- Debe validar que el usuario ingrese datos correctos según corresponda y mostrar el mensaje de error respectivo (ver los casos de prueba).

Recordar que:

Para determinar si dos circunferencias se cruzan debe tener en cuenta lo siguiente:

- Si la suma de los radios de las circunferencias es mayor o igual que la distancia entre sus centros, entonces las circunferencias se cruzan; es decir:

  $$radio_1 + radio_2 >= distancia$$

  donde:

  - $radio_1$ es el radio de la circunferencia 1.
  - $radio_2$ es el radio de la circunferencia 2.
  - $distancia$ es la distancia entre los centros de las circunferencias.

- Para calcular el radio (r) y el centro de la circunferencia (h,k) debe tener en cuenta que:

  - $h = -(C/2)$
  - $k = -(D/2)$
  - $r = \sqrt{h^2 + k^2 - E}$

- La distancia entre dos puntos se calcula mediante:

  $$distancia = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Recordar que:

Dadas dos rectas: $A_1x + B_1y + C_1 = 0$ y $A_2x + B_2y + C_2 = 0$

- Son coincidentes si:

$$\frac{A_1}{A_2} = \frac{B_1}{B_2} = \frac{C_1}{C_2}$$

- Son paralelas si:

$$\frac{A_1}{A_2} = \frac{B_1}{B_2} \neq \frac{C_1}{C_2}$$

- Son secantes si:

$$\frac{A_1}{A_2} \neq \frac{B_1}{B_2}$$

Nota:

En el problema planteado, para comparar dos números decimales considere que son iguales si la diferencia entre ellos es menor o igual a un margen de error establecido de 0,01.

Caso de prueba 1

Ingrese la opcion que desea calcular:
Circunferencias que se intersectan (A o a)
Relación entre rectas (B o b)
R
Error en la opción ingresada.

Caso de prueba 2

Ingrese la opcion que desea calcular:
Circunferencias que se cruzan (A o a)
Relación entre rectas (B o b)
a
Ingrese los coeficientes C, D y E de la circunferencia 1 ($x^2 + y^2 + Cx + Dy + E$):2 -8 13
Ingrese los coeficientes C, D y E de la circunferencia 2 ($x^2 + y^2 + Cx + Dy + E$):-40 8 79
Las circunferencias no se cruzan.

Caso de prueba 3

Ingrese la opcion que desea calcular:
Circunferencias que se cruzan (A o a)
Relación entre rectas (B o b)
A
Ingrese los coeficientes C, D y E de la circunferencia 1 ($x^2 + y^2 + Cx + Dy + E$):10 5 -1
Ingrese los coeficientes C, D y E de la circunferencia 2 ($x^2 + y^2 + Cx + Dy + E$):4 -5 1
Las circunferencias se cruzan

Caso de prueba 4

Ingrese la opcion que desea calcular:
Circunferencias que se intersectan (A o a)
Relación entre rectas (B o b)
b
Ingrese los coeficientes A, B y C de la recta 1 ($Ax + By + C = 0$):62.82 62.94 -11202.84
Ingrese los coeficientes A, B y C de la recta 2 ($Ax + By + C = 0$):-23.42 -30.74 7445.18
Las rectas son secantes.

Caso de prueba 5

Ingrese la opcion que desea calcular:
Circunferencias que se intersectan (A o a)
Relación entre rectas (B o b)
b
Ingrese los coeficientes A, B y C de la recta 1 ($Ax + By + C = 0$):67.82 62.94 -30776.65
Ingrese los coeficientes A, B y C de la recta 2 ($Ax + By + C = 0$):67.82 62.94 -47813.37
Las rectas son paralelas.

Caso de prueba 6

Ingrese la opcion que desea calcular:
Circunferencias que se intersectan (A o a)
Relación entre rectas (B o b)
b
Ingrese los coeficientes A, B y C de la recta 1 ($Ax + By + C = 0$):1 -2 3
Ingrese los coeficientes A, B y C de la recta 2 ($Ax + By + C = 0$):-2 4 -6
Las rectas son coincidentes.

Caso de prueba 7

Ingrese la opcion que desea calcular:
Circunferencias que se intersectan (A o a)
Relación entre rectas (B o b)
b
Ingrese los coeficientes A, B y C de la recta 1 ($Ax + By + C = 0$):2 3 -1
Ingrese los coeficientes A, B y C de la recta 2 ($Ax + By + C = 0$):4 6 -5
Las rectas son paralelas.