

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

ESTUDIOS GENERALES CIENCIAS

1INF01 - FUNDAMENTOS DE PROGRAMACIÓN

Guía de laboratorio #4

Elaboración de programas con estructuras selectivas dobles



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

1 de mayo de 2019

Índice general

Historial de revisiones	1
Siglas	2
1. Guía de Laboratorio #4	3
1.1. Introducción	3
1.2. Materiales y métodos	3
1.3. Estructura Selectiva Doble	3
1.3.1. Representación de la Estructura Selectiva Doble	4
1.3.2. Representación en pseudocódigo	4
1.3.3. Representación en diagrama de flujo	5
1.3.4. Implementación en ANSI C	5
1.4. Cálculo de las raíces ecuaciones cuadráticas con una variable	6
1.5. ¿Cuántos dígitos tiene un número?	8
1.6. Impresión en diferentes bases	10
1.7. Funciones definida por tramos	11
1.8. Calculadora	12
1.9. Ejercicios propuestos	14
1.9.1. Cantidad de días por mes	14
1.9.2. El mayor de 3 números	15
1.9.3. Funciones definida por tramos	16
1.9.4. La Media Aritmética-Geométrica	16
1.9.5. Tipos de rectas	17
1.9.6. Clasificación de un paralelogramo según sus lados	17
1.9.7. Cálculo de la jornada de trabajo semanal	19
1.9.8. Distancia más cercana	19
1.9.9. Comparación de fechas	20
1.9.10. Conversión de coordenadas rectangulares a polares	21
1.9.11. Conversión de grados	21
1.9.12. Conversión de temperatura	22

1.9.13. Aplicación de la ley de Boyle	22
1.9.14. Aplicación de la ley de Charles	23
1.9.15. Calculadora de operaciones lógicas	24
1.9.16. Calculadora de operaciones vectoriales	24
1.9.17. Calculadora de números complejos	25
1.9.18. La escalas de Wechsler	25
1.9.19. Categorización del Índice de Masa Corporal	26
1.9.20. Suma de n número naturales	26
1.9.21. Manipulación de número enteros	27
1.9.22. Cálculo de las raíces ecuaciones cuadráticas con una variable	28

FUNDAMENTOS DE PROGRAMACIÓN
ESTUDIOS GENERALES CIENCIAS
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Historial de Revisiones

Revisión	Fecha	Autor(es)	Descripción
1.0	14.09.2018	A.Melgar	Versión inicial.
1.1	27.04.2019	A.Melgar	Se incrementó la cantidad de problemas propuestos, se añadió color al código en ANSI C y se completaron los casos de prueba de los problemas propuestos.

Siglas

ASCII	American Standard Code for Information Interchange
ANSI	American National Standards Institute
EEGGCC	Estudios Generales Ciencias
IDE	Entorno de Desarrollo Integrado
OMS	Organización Mundial de la Salud
PUCP	Pontificia Universidad Católica del Perú
RAE	Real Academia Española

Capítulo 1

Guía de Laboratorio #4

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 selectivas dobles”.

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 selectivas dobles 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 selectiva dobles.
- Diseñe algoritmos expresados en diagramas de flujo y pseudocódigos que controlen el flujo usando una estructura selectiva doble.
- Implemente programas que utilicen la estructura selectiva dobles en un lenguaje de programación imperativo.
- Implemente programas que utilicen estructuras selectivas dobles anidadas en un lenguaje de programación imperativo.

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 ANSI C. Como Entorno de Desarrollo Integrado (**IDE**) para el lenguaje ANSI C se utilizará **Dev C++**². No obstante, es posible utilizar otros **IDEs** como **Netbeans** y **Eclipse**.

1.3. Estructura Selectiva Doble

Como ya se comentó en la guía #3, los algoritmos siguen un flujo de ejecución el cual puede ser modificado. En el mundo de la programación este flujo se puede modificar a través de estructuras de control de flujo.

¹<http://pseint.sourceforge.net/>

²<http://sourceforge.net/projects/orwelldevcpp>

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 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 determinada condición. Las estructuras algorítmicas selectivas por su lado se pueden clasificar en estructuras selectivas simples y estructuras selectivas dobles. En esta guía se estudiará la estructura selectiva doble.

Recordar que:

La estructura selectiva simple permite ejecutar un conjunto de instrucciones si y solo si se cumple determinada condición. Si la condición no se cumple, el conjunto de instrucciones no se ejecuta.

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 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 conjunto 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 como realizar determinado procesamiento. Muchas situaciones requieren que se realice el cálculo de diferente manera dependiendo de la condición. Por ejemplo para calcular la 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 antes 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 Selectiva Doble

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

1.3.2. Representación en pseudocódigo

En la figura 1.1 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: $i < 100$, $nota \leq 10$, $i \leq max$ 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 conjunto de instrucciones, un conjunto de instrucciones que se ejecuta si se cumple la condición (conjunto de instrucciones a en la figura 1.1) y otro conjunto de instrucciones sino se cumple la condición (conjunto de instrucciones b en la figura 1.1). Para separar ambos conjuntos de instrucciones se utiliza el identificador SiNo.

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

```

Si condicion Entonces
|   conjunto de instrucciones a;
SiNo
|   conjunto de instrucciones b;
Fin Si

```

Figura 1.1: Pseudocódigo: Estructura selectiva doble

1.3.3. Representación en diagrama de flujo

En la figura 1.2 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.

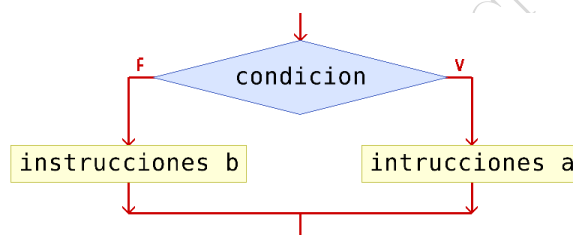


Figura 1.2: Diagrama de Flujo: Selectiva doble

1.3.4. Implementación en ANSI C

En el lenguaje ANSI C la estructura selectiva doble se implementa a través de la instrucción `if`. La representación del `if` se puede apreciar en el programa 1.1. El segundo conjunto de instrucciones se coloca luego del identificador `else`. El conjunto de instrucciones en ANSI C se delimita por los símbolos `{` y `}`. Esta delimitación es opcional cuando el conjunto de instrucciones está formada por una sola instrucción.

Recordar que:

El ANSI C no implementa nativamente el tipo de dato lógico (`bool` o `boolean`). Entonces ¿Cómo hace el ANSI C para evaluar la condición de la instrucción `if`? ANSI 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 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 ≤ 100`, `i == 10`, `suma < i`, `i`. Por otro lado, las siguientes expresiones serán consideradas *falsas*: `suma ≥ 100`, `i == 20`, `suma > i`, `suma`.

Listing 1.1: ANSI C: Estructura selectiva doble

```

1  ...
2  if (condicion){
3      conjunto de instrucciones a;
4  }
5  else{
6      conjunto de instrucciones b;
7  }
8  ...

```


1.4. Cálculo de las raíces ecuaciones cuadráticas con una variable

Una ecuación cuadrática con una variables 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 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 ANSI C que calcule las 2 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 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 guías anteriores. 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 1.3 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 $discriminantes \geq 0$). A diferencia que en la selectiva simple, en este caso existe un conjunto de instrucciones si es que no se cumple la condición.

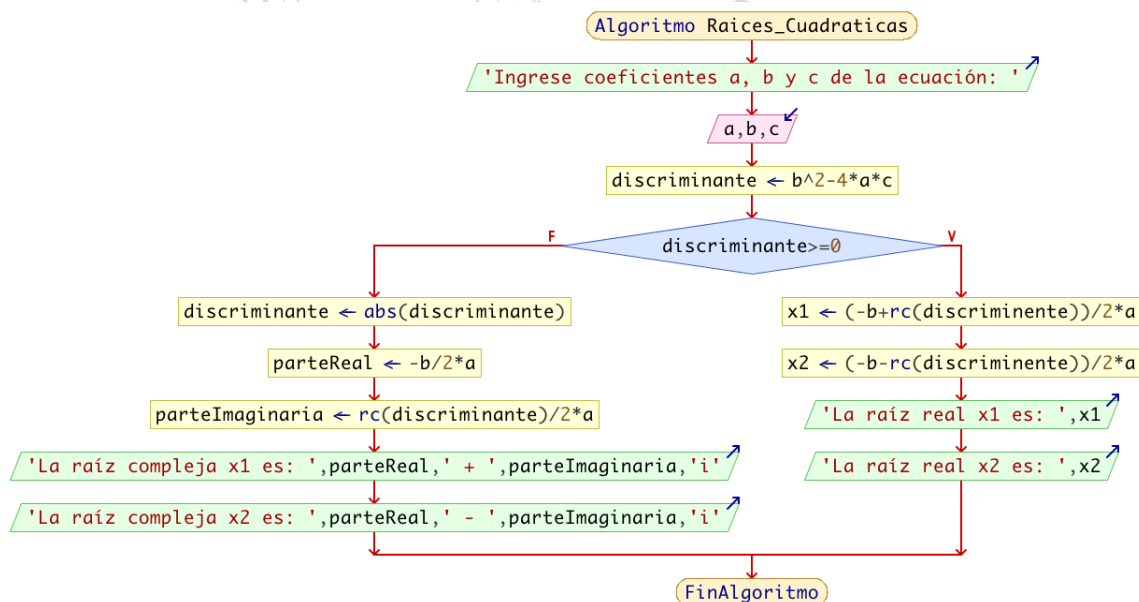


Figura 1.3: Diagrama de flujo: Ecuación cuadrática con raíces reales y complejas

En la figura 1.4 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.

Recordar que:

La indentación es un sangrado 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 ejecuta 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.

```

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       x1 <- (-b+rc(discriminante))/2*a
7       x2 <- (-b-rc(discriminante))/2*a
8       Escribir 'La raíz real x1 es: ',x1
9       Escribir 'La raíz real x2 es: ',x2
10  SiNo
11      discriminante <- abs(discriminante)
12      parteReal <- -b/2*a
13      parteImaginaria <- rc(discriminante)/2*a
14      Escribir 'La raíz compleja x1 es: ',parteReal,' + ',parteImaginaria,'i'
15      Escribir 'La raíz compleja x2 es: ',parteReal,' - ',parteImaginaria,'i'
16  FinSi
17 FinAlgoritmo

```

Figura 1.4: Pseudocódigo: Ecuación cuadrática con raíces reales y complejas

En el programa 1.2 se puede apreciar la alternativa de solución en ANSI 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 se cumple se encuentra entre las líneas 20 – 25. Al igual que en el pseudocódigo, los conjuntos de instrucciones de la selectiva sobre se encuentran indentados.

Recordar que:

Un aspecto importante de la sintaxis del ANSI C es que la expresión que representa a la condición en la estructura selectiva simple y doble siempre va entre paréntesis. Los paréntesis no son necesarios para el caso del diagrama de flujo y pseudocódigo.

Hay dos aspectos importantes a comentar en relación al programa 1.2. El primero vienen en relación a la declaración de las variables `x1`, `x2`, `parteReal` y `parteImaginaria`. Como usted podrá notar en la línea 13 del programa 1.2 se declaran las variables `x1` y `x2` y en la línea 20 del mismo programa, se declaran las variables `parteReal` y `parteImaginaria`. El lenguaje ANSI 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 ANSI C. Como puede notar, en la línea 21 del programa 1.2 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`. Para calcular el valor absoluto de un número entero se utiliza la función `abs` que se encuentra declarada en el archivo de cabecera `stdlib`.

Listing 1.2: Ecuación cuadrática con raíces reales y complejas

```

1  #include <stdio.h>
2  #include <math.h>
3
4  int main() {
5      int a, b, c;
6      double discriminante;
7
8      printf("Ingrese coeficientes a, b y c de la ecuacion: ");
9      scanf("%d %d %d", &a, &b, &c);
10
11     discriminante = pow(b, 2) - 4 * a * c;
12     if (discriminante >= 0) {
13         double x1, x2;
14         x1 = (-b + sqrt(discriminante)) / 2 * a;
15         x2 = (-b - sqrt(discriminante)) / 2 * a;
16         printf("La raiz real x1 es %lf\n", x1);
17         printf("La raiz real x2 es %lf\n", x2);
18     }
19     else{
20         double parteReal, parteImaginaria;
21         discriminante = fabs(discriminante);
22         parteReal = -b/2*a;
23         parteImaginaria = sqrt(discriminante)/2*a;
24         printf("La raiz compleja x1 es %.2lf + %.2lfi\n", parteReal, parteImaginaria);
25         printf("La raiz compleja x2 es %.2lf - %.2lfi\n", parteReal, parteImaginaria);
26     }
27     return 0;
28 }

```

A continuación sigue un ejemplo de ejecución de este programa

Ingrese coeficientes a, b y c de la ecuacion: 1 -4 13

La raiz compleja x1 es 2.00 + 3.00i

La raiz compleja x2 es 2.00 - 3.00i

Para poner en práctica

¿Cuáles son las raíces de las siguientes ecuaciones?

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

1.5. ¿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 selectiva doble. ¿Cuál sería la mejor condición para esta selectiva doble? Luego de analizar el problema, se puede intentar como condición $\text{numero}=0$, 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.

En la figura 1.5 se puede apreciar una propuesta de solución al problema propuesto expresado en diagrama de flujo. En la figura 1.6 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.

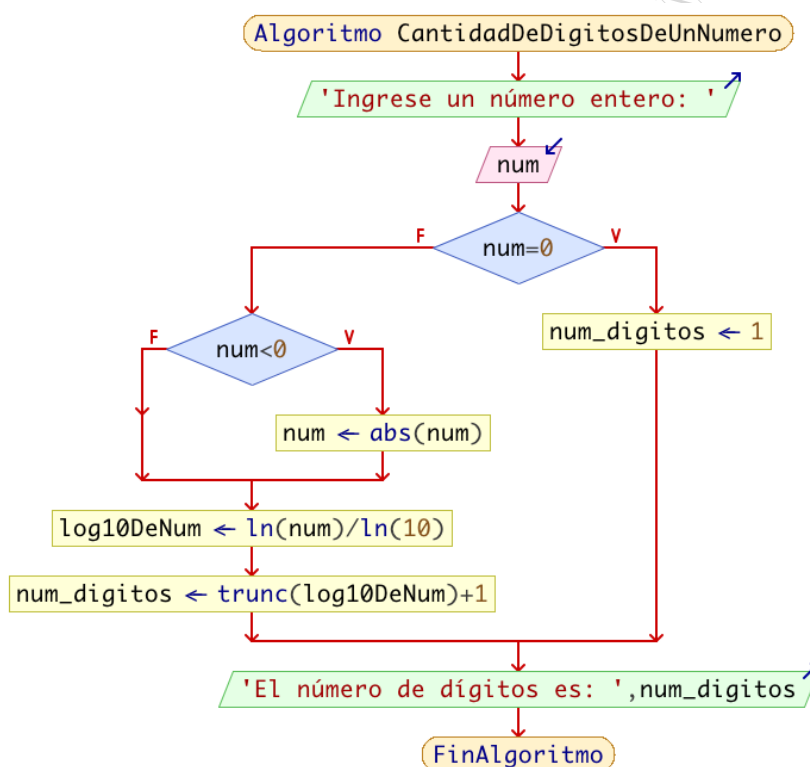


Figura 1.5: Diagrama de flujo: Conteo de dígitos de un número

En el programa 1.3 se puede apreciar la alternativa de solución en un programa en ANSI C. Hay un tres de 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 1.2 en donde se utilizó la función `fabs` cuya declaración se encuentra en el archivo de cabecera `math`. 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`. A diferencia de PSeInt, ANSI C sí define este logaritmo. En ANSI C el logaritmo natural se implementa a través de la función `log`.

El tercer aspecto a considerar es el truncamiento. Existe en ANSI C la función `trunc` pero esta no ha sido empleada en el programa por que al ser la variable `num_digitos` entera, al asignársele un valor real, el ANSI C automáticamente hace un truncamiento.

Listing 1.3: Conteo de dígitos de un número

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

```

1 Algoritmo CantidadDeDigitosDeUnNumero
2   Escribir 'Ingrese un número entero: '
3   Leer num
4   Si num=0 Entonces
5       num_digitos <- 1
6   SiNo
7       Si num<0 Entonces
8           num <- abs(num)
9       FinSi
10      log10DeNum <- ln(num)/ln(10)
11      num_digitos <- trunc(log10DeNum)+1
12  FinSi
13  Escribir 'El número de dígitos es: ',num_digitos
14 FinAlgoritmo

```

Figura 1.6: Pseudocódigo: Conteo de dígitos de un número

```

2 #include <math.h>
3 #include <stdlib.h>
4
5 int main() {
6     int numero, num_digitos;
7
8     printf("Ingrese un numero entero: ");
9     scanf("%d", &numero);
10
11     if (numero==0)
12         num_digitos = 1;
13     else{
14         if (numero<0)
15             numero = abs(numero);
16         num_digitos = log10(numero) + 1;
17     }
18     printf("El numero de digitos es: %d\n", num_digitos);
19     return 0;
20 }

```

1.6. Impresión en diferentes bases

Se desea implementar un programa en ANSI 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 hexadecimal³. Una alternativa de solución a este problema se puede apreciar en el programa 1.4. En este programa, en el conjunto de instrucciones que corresponde cuando no se cumple la condición (líneas 13 – 17), existe otra selectiva doble. Esto se conoce como una estructura selectiva doble anidada.

Listing 1.4: Impresión en diferentes bases

```

1 #include <stdio.h>
2
3 int main() {
4     int numero, base;
5
6     printf("Ingrese numero en base 10: ");
7     scanf("%d", &numero);
8     printf("Ingrese base a transformar (8 o 16): ");

```

³Si se utiliza `%X` los dígitos hexadecimales se imprimirán en mayúscula.

```

9   scanf("%d", &base);
10
11  if (base == 8)
12      printf("El numero en base 8 es %o\n", numero);
13  else
14      if (base == 16)
15          printf("El numero en base 16 es %x\n", numero);
16      else
17          printf("Base no soportada.\n");
18  return 0;
19  }

```

1.7. Funciones definida por tramos

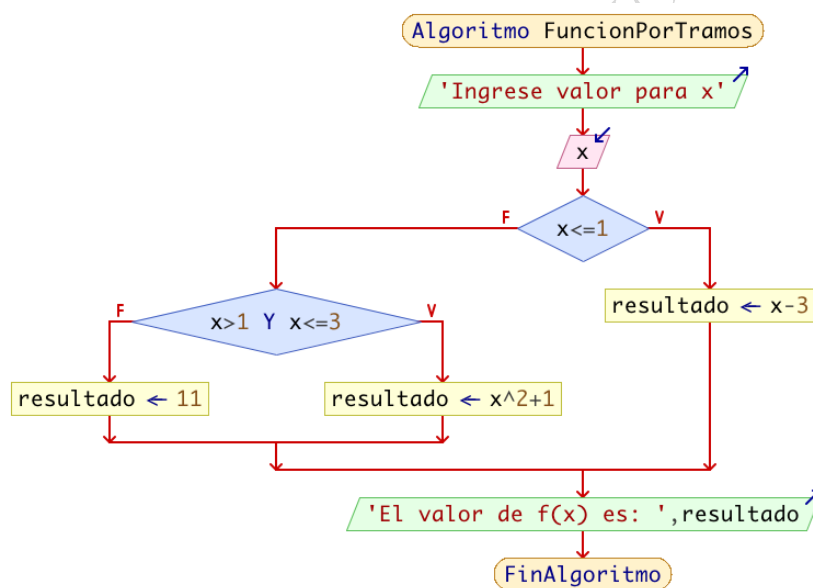


Figura 1.7: Diagrama de flujo: Función definida por tramos

Otra situación en donde se pueden aplicar estructuras selectivas dobles anidadas es en las funciones definidas por tramos como la siguiente:

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

En la figura 1.7 se puede apreciar una alternativa de solución a la función definida por tramos dado determinado número x expresada en diagrama de flujo. En la figura 1.8 se aprecia la misma alternativa de solución expresada en pseudocódigo. La implementación de esta alternativa de solución en ANSI C se puede apreciar en el programa 1.5.

Listing 1.5: Función definida por tramos

```

1  #include <stdio.h>
2  #include <math.h>
3
4  int main() {
5      double x, y;
6
7      printf("Ingrese el valor para x: ");

```

```

1  Algoritmo FuncionPorTramos
2      Escribir 'Ingrese valor para x'
3      Leer x
4      Si x<=1 Entonces
5          resultado <- x-3
6      SiNo
7          Si x>1 y x<=3 Entonces
8              resultado <- x^2+1
9          SiNo
10             resultado <- 11
11         FinSi
12     FinSi
13     Escribir 'El valor de f(x) es: ', resultado
14 FinAlgoritmo

```

Figura 1.8: Pseudocódigo: Función definida por tramos

```

8      scanf("%lf", &x);
9
10     if (x <= 1)
11         y = x-3;
12     else
13         if (x>1 && x<=3)
14             y=pow(x,2)+1;
15         else
16             y=11;
17     printf("El valor de f(x) es %lf\n", y);
18     return 0;
19 }

```

1.8. Calculadora

Otro clásico ejemplo del uso de selectivas dobles anidadas es el de la implementación de una calculadora. Dado dos números que representan a los operandos de una operación y un caracter que representa a una operación (+, -, *, /), se busca que retorne el resultado de aplicar el operador a los operandos. En caso se ingrese una operación no esperada, deberá emitir un mensaje de error.

Una alternativa de solución a este problema utilizando selectivas dobles se puede apreciar en la figura 1.9. El correspondiente pseudocódigo se puede apreciar en la figura 1.10.

La representación del algoritmo se puede apreciar en el programa 1.6. A pesar que este programa aparentemente está bien escrito, posee un error de implementación. Sucede que al leer los operandos el usuario luego de ingresar los números procede a presionar la tecla **enter**. Esta tecla se representa internamente con un caracter especial, y este caracter especial será leído por el `scanf('%c', &operacion)`. Por ese motivo es necesario leer antes ese cambio de línea para que el programa funcione correctamente. Una alternativa de solución se puede apreciar en el programa 1.7.

Listing 1.6: Calculadora de operadores aritméticos

```

1  #include <stdio.h>
2
3  int main() {
4      int numero1, numero2, resultado;
5      char operacion;
6
7      printf("Ingrese dos operandos enteros: ");
8      scanf("%d %d", &numero1, &numero2);
9      printf("Ingrese operacion (+,-,*,/): ");
10     scanf("%c", &operacion);
11 }

```

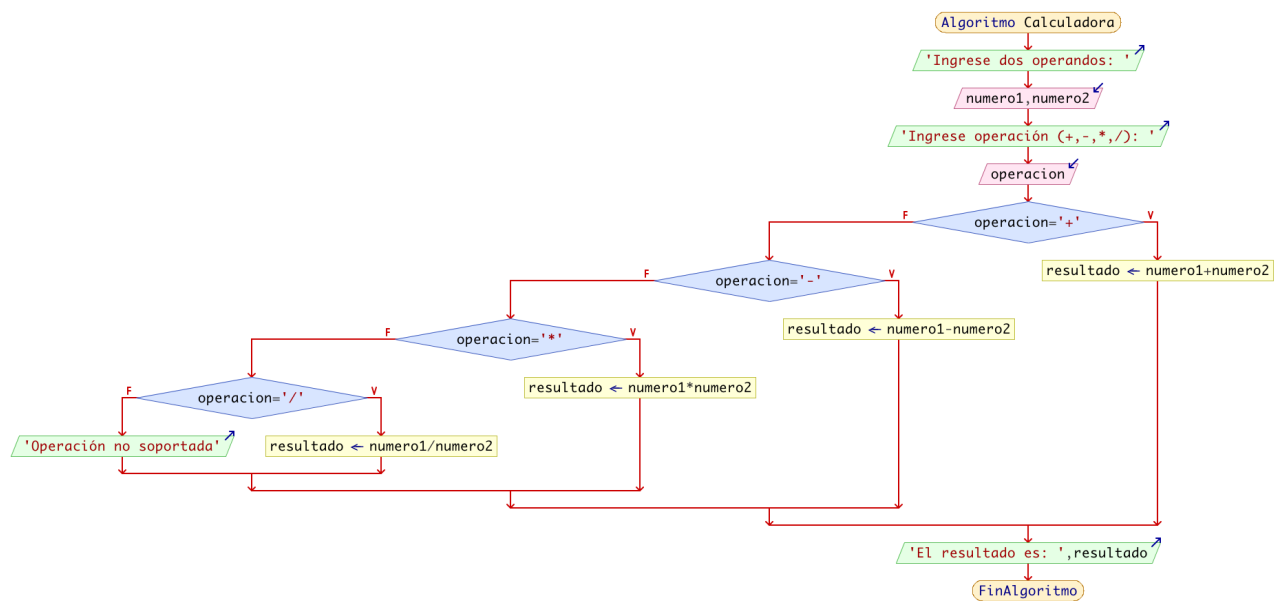


Figura 1.9: Diagrama de flujo: Calculadora de operadores aritméticos

```

1  Algoritmo Calculadora
2  Escribir 'Ingrese dos operandos: '
3  Leer numero1, numero2
4  Escribir 'Ingrese operación (+, -, *, /): '
5  Leer operacion
6  Si operacion = '+' Entonces
7      resultado <- numero1 + numero2
8  SiNo
9      Si operacion = '-' Entonces
10         resultado <- numero1 - numero2
11     SiNo
12         Si operacion = '*' Entonces
13             resultado <- numero1 * numero2
14         SiNo
15             Si operacion = '/' Entonces
16                 resultado <- numero1 / numero2
17             SiNo
18                 Escribir 'Operación no soportada'
19             FinSi
20         FinSi
21     FinSi
22 FinSi
23 Escribir 'El resultado es: ', resultado
24 FinAlgoritmo
--

```

Figura 1.10: Pseudocódigo: Calculadora de operadores aritméticos

```

12  if (operacion=='+')
13      resultado = numero1 + numero2;
14  else if (operacion=='-')
15      resultado = numero1 - numero2;
16  else if (operacion=='*')
17      resultado = numero1 * numero2;
18  else if (operacion=='/')
19      resultado = numero1 / numero2;
20  else

```



```

21     printf("Operacion no soportada\n");
22     printf("El resultado es: %d\n", resultado);
23     return 0;
24 }

```

Listing 1.7: Calculadora de operadores aritméticos

```

1  #include <stdio.h>
2
3  int main() {
4      int numero1, numero2, resultado;
5      char operacion, cambio_de_linea;
6
7      printf("Ingrese dos operandos enteros: ");
8      scanf("%d %d", &numero1, &numero2);
9      printf("Ingrese operacion (+,-,*,/): ");
10     scanf("%c", &cambio_de_linea);
11     scanf("%c", &operacion);
12
13     if (operacion=='+')
14         resultado = numero1 + numero2;
15     else if (operacion=='-')
16         resultado = numero1 - numero2;
17     else if (operacion=='*')
18         resultado = numero1 * numero2;
19     else if (operacion=='/')
20         resultado = numero1 / numero2;
21     else
22         printf("Operacion no soportada\n");
23     printf("El resultado es: %d\n", resultado);
24     return 0;
25 }

```

1.9. Ejercicios propuestos

Para cada uno de los 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 ANSI C conforme a los temas revisados en las guías #1, #2 y #3 del curso Fundamentos de Programación.

1.9.1. 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 proposiciones:

- 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.

1.9.2. 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.

1.9.3. Funciones definida por tramos

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

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

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 ANSI C

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

1.9.4. 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*⁴. En esta escala se definen categorías nominales que, dependiendo del valor de coeficiente intelectual (CI), permiten definir la inteligencia. En la tabla 1.1 se pueden apreciar las escalas nominales de esta prueba de inteligencia.

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

1.9.5. 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

⁴https://comenio.files.wordpress.com/2007/09/weshler_imprimir.pdf

Tabla 1.1: 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

dividiendo el peso de una persona en kilos por el cuadrado de su talla en metros (kg/m^2)⁵. La Organización Mundial de la Salud (OMS) clasifica el estado nutricional utilizando el IMC conforme la tabla 1.2.

Tabla 1.2: 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.

1.9.6. 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_0 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.

⁵<http://www.who.int/es/news-room/fact-sheets/detail/obesity-and-overweight>

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.

1.9.7. 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.

Recordar 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 $m = \text{indefinida}$ 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**.

1.9.8. 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 categorías (ver figura 1.11).

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 caracteriza 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 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.

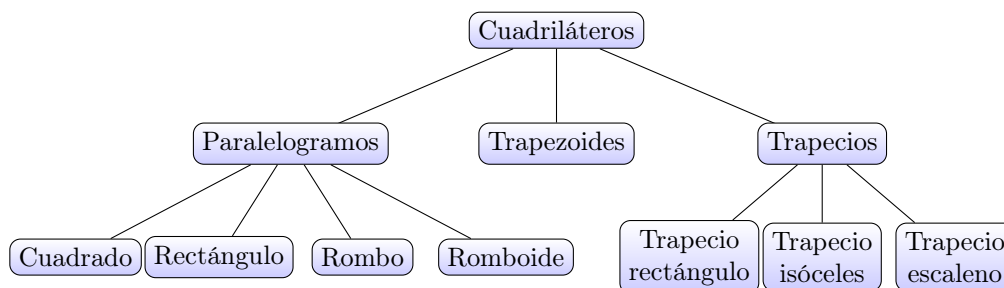


Figura 1.11: Clasificación de cuadriláteros

Dentro de los Trapecios, el Cuadrilátero se puede clasificar como Trapecio rectángulo, Trapecio isósceles y Trapecio escaleno. El Trapecio rectángulo se caracteriza por tener 2 ángulos rectos. El Trapecio isósceles se caracteriza 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 cuadrilátero, determine el posible paralelogramo en el cual se podría clasificar.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si $l_1 = 4, l_2 = 4, l_3 = 4, l_4 = 4$, entonces se debe imprimir Posible Cuadrado o Rombo.
- Si $l_1 = 4, l_2 = 6, l_3 = 6, l_4 = 4$, entonces se debe imprimir Posible Rectángulo o Romboide.
- Si $l_1 = 6, l_2 = 4, l_3 = 6, l_4 = 4$, entonces se debe imprimir Posible Rectángulo o Romboide.
- Si $l_1 = 3, l_2 = 3, l_3 = 5, l_4 = 5$, 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.

1.9.9. 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.

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.

1.9.10. 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.

Recordar 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.

1.9.11. 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 igual. 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.

Recordar 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.

1.9.12. 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:

$$r = \sqrt{x^2 + y^2}$$

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

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 ANSI C

- En ANSI 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`. El resultado de esta función se retorna en radianes.

1.9.13. 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.

1.9.14. 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.

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

1.9.16. 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 $^{\circ}$.

Casos de prueba

Use los siguientes datos para probar su solución:

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

1.9.17. 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 $\text{operación} = C$ se debe imprimir El resultado es V.
- Si $p = V$, $q = F$ y $\text{operación} = C$ se debe imprimir El resultado es F.
- Si $p = F$, $q = F$ y $\text{operación} = D$ se debe imprimir El resultado es F.
- Si $p = V$, $q = F$ y $\text{operación} = D$ se debe imprimir El resultado es V.
- Si $p = V$, $q = F$ y $\text{operación} = K$ se debe imprimir El resultado es F.
- Si $p = F$, $q = F$ y $\text{operación} = B$ se debe imprimir El resultado es V.

1.9.18. Calculadora de operaciones vectoriales

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

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

Casos de prueba

Utilice los siguientes datos para probar su solución.

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

1.9.19. 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., $a + bi$).

1.9.20. Suma de n número 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 $x > 0$ e $y \in [1..3]$.

Recordar que:

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

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

Recordar que:

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

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

Recordar que:

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

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

1.9.21. 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

1.9.22. 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 $|z|(\cos(\theta) + i\sin(\theta))$

Si se tiene el número de la forma $x + yi$, se puede calcular:

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

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 [1.9.21](#).