

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

ESTUDIOS GENERALES CIENCIAS

1INF01 - FUNDAMENTOS DE PROGRAMACIÓN

Guía de laboratorio #5

Elaboración de programas con estructuras selectivas anidadas



8 de septiembre de 2021

Índice general

Historial de revisiones	1
Siglas	2
1. Guía de Laboratorio #5	3
1.1. Introducción	3
1.2. Materiales y métodos	3
1.3. Estructuras Algorítmicas Selectivas Anidadas	3
1.3.1. Representación de la Estructura Selectiva Anidada	4
1.3.2. Representación en pseudocódigo	4
1.3.3. Representación en diagrama de flujo	5
1.3.4. Implementación en C	5
1.4. Cálculo de las raíces ecuaciones cuadráticas con una variable	6
1.5. ¿Cuántos dígitos tiene un número?	10
1.6. Impresión en diferentes bases	12
1.7. Funciones definida por tramos	13
1.8. Calculadora	14
2. Ejercicios propuestos	17
2.1. Nivel básico	17
2.1.1. El mayor de 3 números	17
2.1.2. Cantidad de días por mes	17
2.1.3. Calculadora de operaciones lógicas	18
2.1.4. Conversión de grados	19
2.1.5. Conversión de temperatura	19
2.1.6. Aplicación de la ley de Boyle	20
2.1.7. Aplicación de la ley de Charles	21
2.1.8. La escalas de Wechsler	21
2.1.9. Cálculo de la jornada de trabajo semanal	22
2.2. Nivel intermedio	22

2.2.1.	Funciones definida por tramos	22
2.2.2.	Categorización del Índice de Masa Corporal	23
2.2.3.	La Media Aritmética-Geométrica	23
2.2.4.	Calculadora de operaciones vectoriales	24
2.2.5.	Distancia más cercana	24
2.2.6.	Calculadora de números complejos	25
2.2.7.	Tipos de rectas	26
2.3.	Nivel avanzado	26
2.3.1.	Clasificación de un paralelogramo según sus lados	26
2.3.2.	Comparación de fechas	28
2.3.3.	Conversión de coordenadas rectangulares a polares	28
2.3.4.	Suma de n número naturales	29
2.3.5.	Manipulación de número enteros	30
2.3.6.	Cálculo de las raíces ecuaciones cuadráticas con una variable	30
2.3.7.	Mayor número posible en un rango	31
2.3.8.	Conteo de vocales y consonantes	32
2.3.9.	Número económico	33
2.3.10.	Suma de Riemann (adaptado del laboratorio 5 del ciclo 2020-2)	35
2.3.11.	Husos horarios versión 1 (adaptado del laboratorio 5 del ciclo 2020-2)	36
2.3.12.	Husos horarios versión 2 (adaptado del laboratorio 5 del ciclo 2020-2)	38
2.3.13.	Números centralizados (adaptado del laboratorio 5 del ciclo 2020-2)	41
2.3.14.	Identificar patrón (adaptado del laboratorio 5 del ciclo 2020-2)	43
2.3.15.	Encriptar o desencriptar un mensaje escondido (adaptado del laboratorio 5 del ciclo 2020-2)	45
2.3.16.	Identificar la cantidad de dígitos (adaptado del laboratorio 5 del ciclo 2020-2)	47
2.3.17.	Condensador (adaptado del laboratorio 5 del ciclo 2020-2)	50
2.3.18.	La resistencia (adaptado del laboratorio 5 del ciclo 2020-2)	53
2.3.19.	Colores en HTML versión 1 (adaptado del laboratorio 5 del ciclo 2020-2)	56
2.3.20.	Colores en HTML versión 2 (adaptado del laboratorio 5 del ciclo 2020-2)	57
2.3.21.	Colores en HTML versión 3 (adaptado del laboratorio 5 del ciclo 2020-2)	58
2.3.22.	Colores en HTML versión 4 (adaptado del laboratorio 5 del ciclo 2020-2)	60
2.3.23.	Identificar Cónicas (adaptado del laboratorio 5 del ciclo 2021-1)	61
2.3.24.	Las capas de la esfera (adaptado del laboratorio 5 del ciclo 2021-1)	65
2.3.25.	Las sumatorias (adaptado del laboratorio 5 del ciclo 2021-1)	67
2.3.26.	Más sumatorias (adaptado del laboratorio 5 del ciclo 2021-1)	69
2.3.27.	La parábola (adaptado del laboratorio 5 del ciclo 2021-1)	71

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 C y se completaron los casos de prueba de los problemas propuestos.
1.1	15.05.2019	L. Hirsh	Revisión de la versión 1.1.
2.0	31.07.2019	A.Melgar	Se cambio el contenido del documento para reflejar el cambio de tema. En este documento ahora se trabajará la estructura algorítmica selectiva anidada.
3.0	22.08.2020	L. Hirsh	Se incluyen nuevos ejercicios y se hace revisión del material.
3.1	01.10.2020	L. Hirsh	Se incluyen nuevos ejercicios, se hace revisión del material y reorganización.
3.2	16.05.2021	I. Caldas	Se incluyen nuevos ejercicios.
3.3	08.09.2021	J. Berrocal	Se incluyen nuevos ejercicios.

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 #5

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 anidadas”.

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 anidadas 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 funcionamiento de las estructuras de control de flujo en particular la utilización de estructuras selectivas anidadas.
- Diseñe algoritmos expresados en diagramas de flujo y pseudocódigos que controlen el flujo usando estructuras selectivas anidadas.
- Implemente programas que utilicen estructuras selectivas 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 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. Estructuras Algorítmicas Selectivas Anidadas

Como ya se comentó en la guía #2, 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. Las estructuras de control de flujo pueden ser de dos tipos: estructuras algorítmicas selectivas y estructuras algorítmicas iterativas.

¹<http://pseint.sourceforge.net/>

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

Las estructuras algorítmicas selectivas, permiten que los programas ejecuten un conjunto de instrucciones si es que se cumple determinada condición. Es decir, el conjunto de instrucciones se ejecuta una sola vez. 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.

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.

Recordar que:

La estructura selectiva doble permite ejecutar un conjunto de instrucciones si se cumple determinada condición. Si la condición no se cumple, se ejecuta otro conjunto de instrucciones. Es decir, con la estructura selectiva doble se ejecuta solamente uno de los dos conjuntos de instrucciones definidos en la estructura, pero siempre se ejecuta uno de ellos.

Con la selectiva simple se puede tomar la decisión de ejecutar o no un conjunto de instrucciones. Con la selectiva doble se puede tomar la decisión de ejecución sobre dos conjuntos de instrucciones, se ejecuta solo uno de ellos. Cada conjunto de instrucciones puede ser visto como una alternativa que el algoritmo puede escoger.

La limitación que trae la estructura algorítmica selectiva, tal como se ha estudiado en la guía #2, 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 discriminante 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.

1.3.1. 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.

1.3.2. Representación en pseudocódigo

En la figura 1.1 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*.

```

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 1.1: Pseudocódigo: Estructuras selectivas anidadas

1.3.3. Representación en diagrama de flujo

En la figura 1.2 se puede apreciar la representación en diagrama flujo correspondiente al pseudocódigo presentado en la figura 1.1.

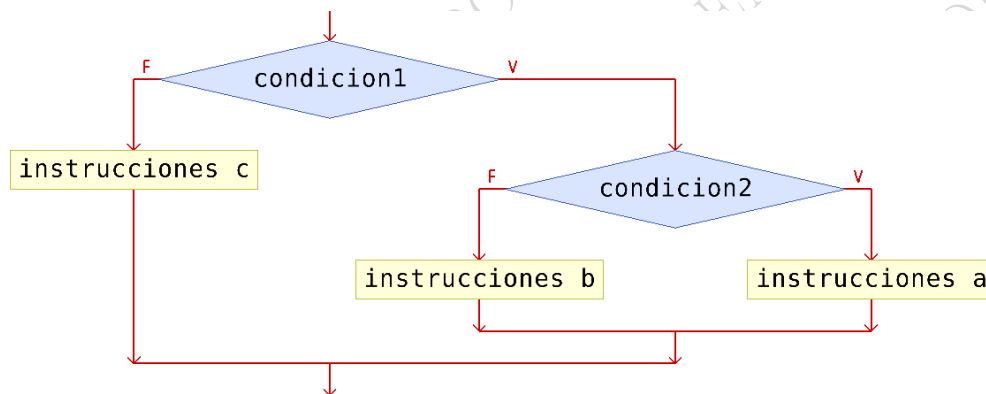


Figura 1.2: Diagrama de Flujo: Selectiva anidada

1.3.4. Implementación en C

En el programa 1.1 se puede apreciar el mismo ejemplo presentado en las figuras 1.2 y 1.1 en C.

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 ≤ 100`, `i == 10`, `suma < i`, `i`. Por otro lado, las siguientes expresiones serán consideradas *falsas*: `suma ≥ 100`, `i == 20`, `suma > i`, `suma`.

Programa 1.1: C: Estructura selectiva anidada

```

1  ...
2  if (condición1){
3      if (condición2){
4          conjunto de instrucciones a;
5      }
6      else{
7          conjunto de instrucciones b;
8      }
9  }
10 else{
11     conjunto de instrucciones c;
12 }
13 ...

```

1.4. Cálculo de las raíces ecuaciones cuadráticas con una variable

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 discriminante 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 1.3 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 $\text{discriminante} \geq 0$, incluye otra estructura selectiva doble cuya condición es $\text{discriminante} = 0$.

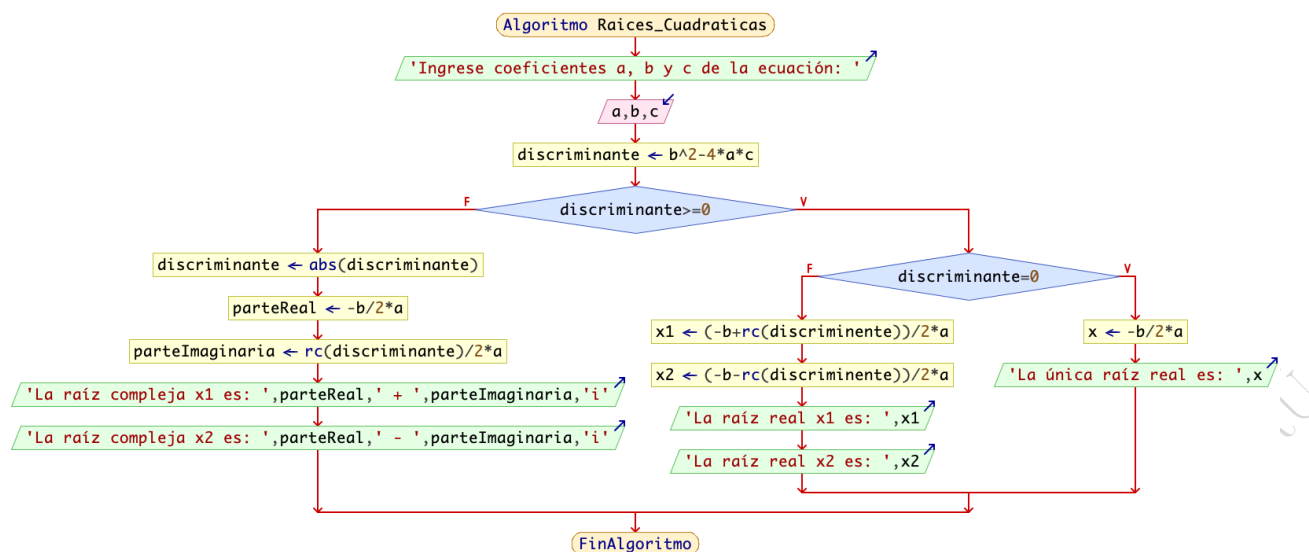


Figura 1.3: Diagrama de flujo: Ecuación cuadrática con raíces reales y complejas

En la figura 1.4 se aprecia el pseudocódigo de la misma alternativa de solución. 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.

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.

En el programa 1.2 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. En el programa 1.3 se muestra la misma versión pero usando funciones.

```

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 discriminante=0 Entonces
7              x <- -b/2*a
8              Escribir 'La única raíz real es: ',x
9          SiNo
10             x1 <- (-b+rc(discriminante))/2*a
11             x2 <- (-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 <- -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 1.4: Pseudocódigo: Ecuación cuadrática con raíces reales y complejas

Programa 1.2: Ecuación cuadrática con raíces reales y complejas

```

1  #include <stdio.h>
2  #include <math.h>
3
4  int main() {
5      double a, b, c;
6      double discriminante;
7
8      printf("Ingrese coeficientes a, b y c de la ecuación: ");
9      scanf("%lf %lf %lf", &a, &b, &c);
10
11     discriminante = pow(b, 2) - 4 * a*c;
12     if (discriminante >= 0) {
13         if (discriminante == 0) {
14             double x;
15             x = (-b + sqrt(discriminante)) / 2 * a;
16             printf("La única raíz real es %lf\n", x);
17         } else {
18             double x1, x2;
19             x1 = (-b + sqrt(discriminante)) / 2 * a;
20             x2 = (-b - sqrt(discriminante)) / 2 * a;
21             printf("La raíz real x1 es %lf\n", x1);
22             printf("La raíz real x2 es %lf\n", x2);
23         }
24     } else {
25         double parteReal, parteImaginaria;
26         discriminante = fabs(discriminante);
27         parteReal = -b / 2 * a;
28         parteImaginaria = sqrt(discriminante) / 2 * a;
29         printf("La raíz compleja x1 es %.2lf + %.2lfi\n", parteReal, parteImaginaria);
30         printf("La raíz compleja x2 es %.2lf - %.2lfi\n", parteReal, parteImaginaria);
31     }
32     return 0;
33 }

```

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?

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

Programa 1.3: Ecuación cuadrática con raíces reales y complejas

```

1  #include <stdio.h>
2  #include <math.h>
3
4  void muestra_raiz_unica(double discriminante, double a, double b);
5  void muestra_raices_reales(double discriminante, double a, double b);
6  void muestra_raices_complejas(double discriminante, double a, double b);
7
8  int main() {
9      double a, b, c;
10     double discriminante;
11
12     printf("Ingrese coeficientes a, b y c de la ecuación: ");
13     scanf("%lf %lf %lf", &a, &b, &c);
14
15     discriminante = pow(b, 2) - 4 * a * c;
16     if (discriminante >= 0) {
17         if (discriminante == 0)
18             muestra_raiz_unica(discriminante, a, b);
19         else
20             muestra_raices_reales(discriminante, a, b);
21     } else
22         muestra_raices_complejas(discriminante, a, b);
23     return 0;
24 }
25
26 void muestra_raiz_unica(double discriminante, double a, double b) {
27     double x;
28     x = (-b + sqrt(discriminante)) / 2 * a;
29     printf("La única raíz real es %lf\n", x);
30 }
31
32 void muestra_raices_reales(double discriminante, double a, double b) {
33     double x1, x2;
34     x1 = (-b + sqrt(discriminante)) / 2 * a;
35     x2 = (-b - sqrt(discriminante)) / 2 * a;
36     printf("La raíz real x1 es %lf\n", x1);
37     printf("La raíz real x2 es %lf\n", x2);
38 }
39
40 void muestra_raices_complejas(double discriminante, double a, double b) {
41     double parteReal, parteImaginaria;
42     discriminante = fabs(discriminante);
43     parteReal = -b / 2 * a;
44     parteImaginaria = sqrt(discriminante) / 2 * a;
45     printf("La raíz compleja x1 es %2lf + %2lfi\n", parteReal, parteImaginaria);
46     printf("La raíz compleja x2 es %2lf - %2lfi\n", parteReal, parteImaginaria);
47 }

```

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 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 $\text{numero}=0$, 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. Esto lo realizaremos con la estructura algorítmica selectiva anidada.

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.

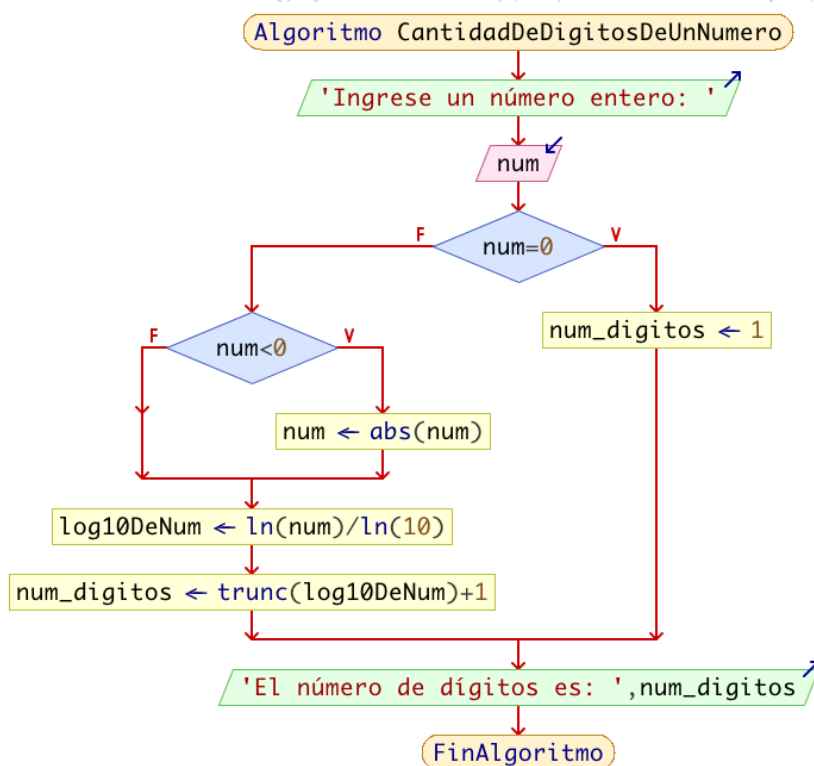


Figura 1.5: Diagrama de flujo: Conteo de dígitos de un número

En el programa 1.4 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

```

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

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

Programa 1.4: Conteo de dígitos de un número

```

1  #include <stdio.h>
2  #include <math.h>
3  #include <stdlib.h>
4
5  int main() {
6      int numero, num_digitos;
7
8      printf("Ingrese un número 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 número de dígitos es: %d\n", num_digitos);
19     return 0;
20 }

```

En el programa 1.5 se aprecia una versión del programa 1.4 en donde el conteo de dígito se realiza en una función definida por el usuario.

Programa 1.5: Conteo de dígitos de un número

```

1  #include <stdio.h>
2  #include <math.h>
3  #include <stdlib.h>
4
5  int contar_digitos(int numero);
6
7  int main() {
8      int numero, num_digitos;
9
10     printf("Ingrese un número entero: ");
11     scanf("%d", &numero);
12
13     num_digitos = contar_digitos(numero);
14     printf("El número de dígitos es: %d\n", num_digitos);
15     return 0;
16 }
17
18 int contar_digitos(int numero) {
19     if (numero == 0)
20         return 1;
21     else {
22         if (numero < 0)
23             numero = abs(numero);
24         return log10(numero) + 1;
25     }
26 }

```

1.6. 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 hexadecimal³. Una alternativa de solución a este problema se puede apreciar en el programa 1.6. 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 1.6: Impresión en diferentes bases

```

1  #include <stdio.h>
2
3  int main() {
4      int numero, base;
5
6      printf("Ingrese número en base 10: ");
7      scanf("%d", &numero);
8      printf("Ingrese base a transformar (8 o 16): ");
9      scanf("%d", &base);
10
11     if (base == 8)
12         printf("El número en base 8 es %o\n", numero);
13     else
14         if (base == 16)
15             printf("El número en base 16 es %x\n", numero);
16         else
17             printf("Base no soportada.\n");
18     return 0;
19 }

```

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

1.7. 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 $x \leq 1$, 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 1.7 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 1.8 se aprecia la misma solución en pseudocódigo. La implementación de esta alternativa de solución en C se puede apreciar en el programa 1.7.

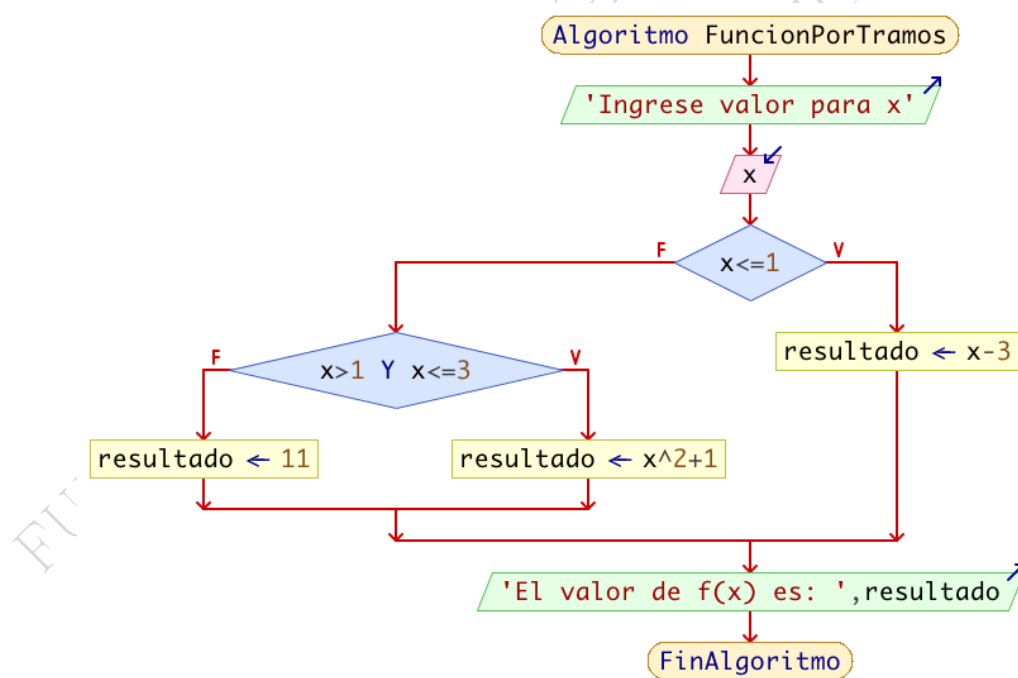


Figura 1.7: Diagrama de flujo: Función definida por tramos


```

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

Programa 1.7: 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: ");
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 anidadas se puede apreciar en la figura 1.9. El correspondiente pseudocódigo se puede apreciar en la figura 1.10.

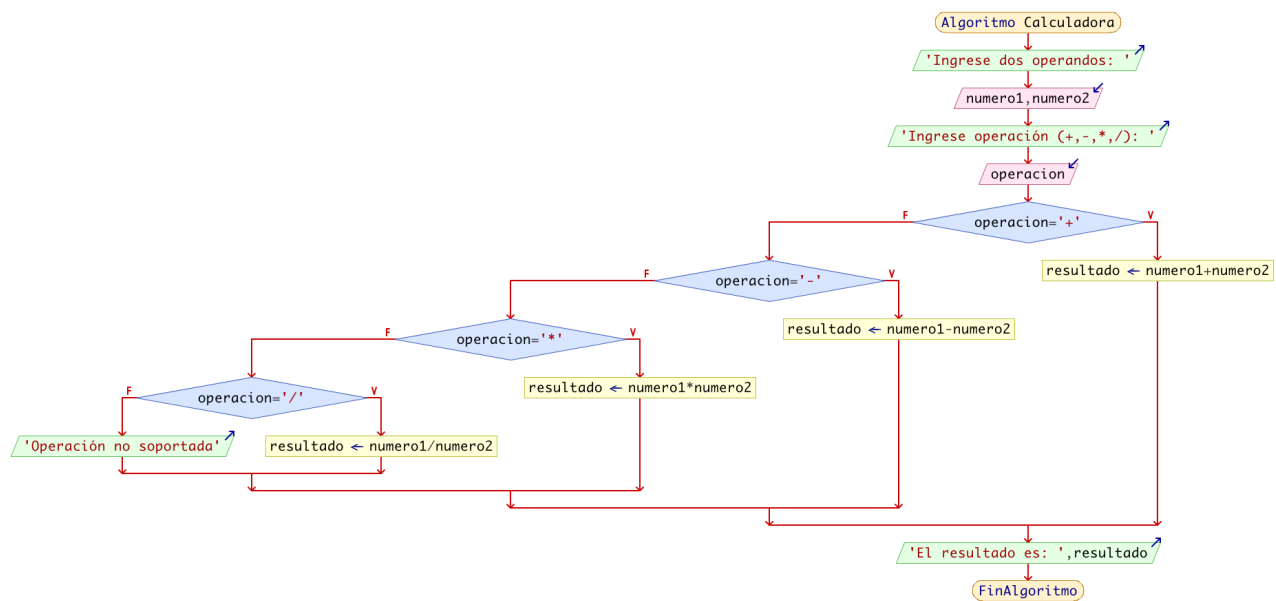


Figura 1.9: Diagrama de flujo: Calculadora de operadores aritméticos

```

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

La representación del algoritmo se puede apreciar en el programa 1.8. A pesar de 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 carácter especial, y este carácter 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 1.9.

Programa 1.8: 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 operación (+,-,*,/): ");
10    scanf("%c", &operacion);
11
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("Operación no soportada\n");
22    printf("El resultado es: %d\n", resultado);
23    return 0;
24 }
```

Programa 1.9: 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 operación (+,-,*,/): ");
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("Operación no soportada\n");
23    printf("El resultado es: %d\n", resultado);
24    return 0;
25 }
```

Capítulo 2

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 C conforme a los temas revisados en las guías preliminar, #1, #2, #3 y #4 del curso Fundamentos de Programación. Se recomienda que las soluciones en C incluyan funciones definidas por el usuario.

2.1. Nivel básico

2.1.1. 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.

2.1.2. 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.

2.1.3. 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.

2.1.4. 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.

2.1.5. 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.

2.1.6. 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.

2.1.7. 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 $T_1 = 20^\circ\text{C}$ y $T_2 = 90^\circ\text{C}$ 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 $V_1 = 100\text{ cm}^3$, $T_2 = 75^\circ\text{C}$ 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 $V_1 = 300\text{ cm}^3$, $T_1 = 80^\circ\text{C}$ y $T_2 = 35^\circ\text{C}$, entonces deberá imprimir El valor de V_2 es 131.25.

2.1.8. 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.

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

2.1.9. 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.

2.2. Nivel intermedio

2.2.1. 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 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.h`.
- 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.

2.2.2. 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/m^2)². La Organización Mundial de la Salud (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.

2.2.3. 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:

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

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

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

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.

2.2.4. 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* = \cdot se debe imprimir El resultado es $u \cdot v=15$.

2.2.5. 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.

2.2.6. 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: $\frac{a + bi}{c + di} = \frac{ac + bd}{c^2 + d^2} + \left(\frac{bc - ad}{c^2 + d^2}\right)i$

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., $a + bi$).

2.2.7. 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**.

2.3. Nivel avanzado

2.3.1. 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 2.1).

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.

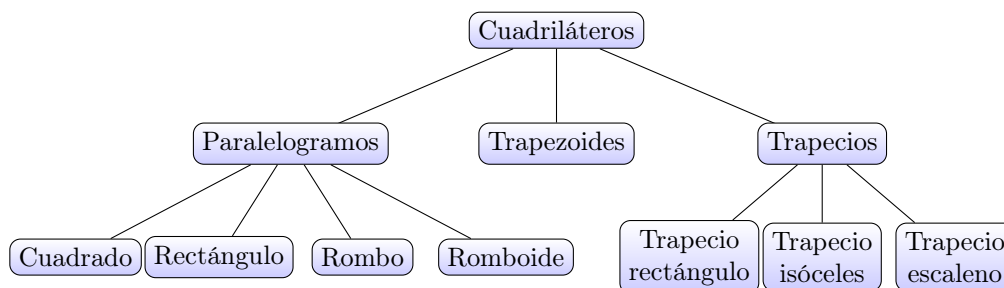


Figura 2.1: Clasificación de cuadriláteros

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.

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 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.

2.3.2. 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.

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.

2.3.3. 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 PSeInt el 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.

2.3.4. 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:

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

2.3.5. 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

2.3.6. 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 2.3.5.

2.3.7. Mayor número posible en un rango

Un número se forma por diversas cifras, las cuales al ser reordenadas pueden generar nuevos números. Por ejemplo, para el número 2494, las cifras a evaluar son 2, 4, 9, 4, con estas cuatro cifras podemos crear el número 4249, 2449 entre otros, pero el máximo número posible es 9442, y el mínimo número posible es el 2449. Se le pide implementar un programa en lenguaje C que solicite al usuario un rango de números y que para cada número de ese rango identifique el máximo número que se puede crear. Para cada número de dicho rango (incluyendo los límites inferior y superior) su programa deberá identificar el máximo número posible y mostrar el número original el número identificado.

Su programa deberá validar lo siguiente:

1. Los valores ingresados por el usuario deben ser positivos, de no serlo debe mostrar el mensaje de error adecuado según sea el caso (El límite inferior no puede ser negativo, El límite superior no puede ser negativo).
2. En caso ambos límites sean positivos, deberá validar que tengan exactamente 4 cifras. Si esta condición no se cumple, deberá imprimir el mensaje correspondiente: El límite inferior debe tener 4 dígitos o El límite superior debe tener 4 dígitos
3. En caso ambos límites sean válidos, deberá verificar que el límite superior sea mayor o igual al límite inferior. Si el usuario ingresa un rango que no cumpla este criterio, su programa deberá imprimir el mensaje de error correspondiente El límite inferior debe ser menor al límite superior.

Recuerde utilizar funciones en su implementación.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Si límite inferior = -1 límite superior = 4, entonces debe imprimir:
El intervalo inferior debe ser positivo.
- Si límite inferior = 123 límite superior = 456, entonces debe imprimir:
El límite inferior debe tener 4 dígitos.
El límite superior debe tener 4 dígitos.
- Si límite inferior = 2019 límite superior = 2021, entonces debe imprimir:
Para 2019 el maximo valor es 9210.
Para 2020 el maximo valor es 2200.
Para 2021 el maximo valor es 2210.
- Si límite inferior = -2019 límite superior = -2024, entonces debe imprimir:
El límite inferior debe ser positivo.
El límite superior debe ser positivo.

Variantes

- En lugar de hallar el mayor halle el menor número.
- Identifique ambos, el mayor y el menor número e imprímalos.

Casos de prueba para variante 1

Utilice los siguientes datos para probar su solución.

- Si límite inferior = 2019 límite superior = 2021, entonces debe imprimir:
Para 2019 el mínimo valor es 129.
Para 2020 el mínimo valor es 22.
Para 2021 el mínimo valor es 122.

Casos de prueba para variante 2

Utilice los siguientes datos para probar su solución.

- Si límite inferior = 2019 límite superior = 2021, entonces debe imprimir:
Para 2019 el maximo valor es 9210 y el mínimo valor es 129.
Para 2020 el maximo valor es 2200 y el mínimo valor es 22.
Para 2021 el maximo valor es 2210 y el mínimo valor es 122.

2.3.8. Conteo de vocales y consonantes

Una palabra puede contener vocales ya sean abiertas(a,e, o) o cerradas(i,u) y consonantes. Se le pide implementar en lenguaje C, un programa que solicite al usuario una serie de letras que formarán una palabra cualquiera y se identifique la cantidad de consonantes, vocales, vocales abiertas y vocales cerradas ingresadas. Su programa debe validar que la cantidad ingresada por el usuario sea mayor a 0 y mostrar el mensaje de error correspondiente de no serlo. En caso de ser válido debe proceder a solicitar al usuario caracter por caracter y verificar que el caracter ingresado sea una letra del alfabeto (A-Z a-z), de no serlo deberá detener la ejecución y mostrar el mensaje de error. Recuerde implementar funciones para su propuesta de solución.

Recordar que:

Las letras del alfabeto poseen un código ASCII.

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Ingrese la cantidad de letras en su palabra: 7
 ingrese la letra 1 de su palabra: a
 ingrese la letra 2 de su palabra: b
 ingrese la letra 3 de su palabra: i
 ingrese la letra 4 de su palabra: e
 ingrese la letra 5 de su palabra: r
 ingrese la letra 6 de su palabra: t
 ingrese la letra 7 de su palabra: o
- vocales: 4 abiertas: 3 cerrada: 1 consonantes: 3

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Ingrese la cantidad de letras en su palabra: 4
 ingrese la letra 1 de su palabra: a
 ingrese la letra 2 de su palabra: #
 ERROR: Debe ingresar letras del alfabeto

Casos de prueba

Utilice los siguientes datos para probar su solución.

- Ingrese la cantidad de letras en su palabra: -6
 ERROR: Debe ingresar una cantidad mayor a 0

2.3.9. Número económico

Un número n se llama económico si el número de dígitos en su factorización de primos (incluyendo exponentes mayores a 1) no es mayor que el número de dígitos de n . En la práctica, los números económicos son la unión de los números equidigitales y frugales. Por ejemplo, el número 128 es frugal en base 10 porque es escrito usando tres dígitos, mientras que su factorización de primos 2^7 usados dígitos. Por ejemplo 144 no es frugal ya que tiene tres dígitos y su factorización de primos $2^4 \cdot 3^2$ usados cinco dígitos.

Se le pide que haciendo uso de lenguaje C elabore un programa que permita saber si un número es equidigital, frugal o económico. Para ello debe, utilizando módulos, poder: identificar si un número es primo o no, identificar cual es el exponente al que se encuentra cierto divisor de un número (por ejemplo el 9 tendría para el divisor 3 un exponente de 2, el 6 tendría para el divisor 2 un exponente de 1 y para el divisor 3 un exponente de 1) y a la vez identificar cuantos dígitos tiene ese exponente (recuerde que el 1 se considera como 0 dígitos), identificar la cantidad de dígitos que posee un número (ojo, no podrá usar fórmula alguna para identificar esta cantidad, deberá utilizar una estructura iterativa para ello). Su programa principal deberá, leer un número a evaluar, validar que sea positivo mayor a 0, identificar la cantidad de dígitos usados en su factorización de primos (por ejemplo para el número 128 esta cantidad sería 2), identificar la cantidad de dígitos que posee el número (por ejemplo para el número 128 sería 3) y finalmente que se muestre que tipo de número es o no es, de tal forma, el número podría ser Frugal, podría ser Equidigital o podría no ser Económico (El 128 sería Frugal ya que $3 < 2$). Para realizar estas acciones debe utilizar los módulos creados.

Casos de prueba para verificación de solución

Ingrese el número de iteraciones: 3
Ingrese el coeficiente y el exponente: 5 4
iteración 1 : suma: 5
iteración 2 : suma: 85
iteración 3 : suma: 490
descartado rango 1
rango 2 - Rpta: [100,1000[

Casos de prueba para verificación de solución

Ingrese el número de iteraciones: 10
Ingrese el coeficiente y el exponente: 6 7
iteración 1 : suma: 6
iteración 2 : suma: 774
iteración 3 : suma: 13896
iteración 4 : suma: 112200
iteración 5 : suma: 580950
iteración 6 : suma: 2260566
iteración 7 : suma: 7201824
iteración 8 : suma: 19784736
iteración 9 : suma: 48482550
iteración 10 : suma: 108482550
descartados rangos 1,2,3 y 4
rango 5 - Rpta: Ninguna de las anteriores

Casos de prueba para verificación de solución

Ingrese el número de iteraciones: 10
Ingrese el coeficiente y el exponente: 12374 11
iteración 1 : suma: 12374
iteración 2 : suma: 25354326
iteración 3 : suma: -2147483648
Se desbordó, el tipo de dato int no es suficientemente grande

Casos de prueba para verificación de solución

Ingrese el número de iteraciones: 4
Ingrese el coeficiente y el exponente: -5 0
ERROR 1:el coeficiente es menor igual a 0

Casos de prueba para verificación de solución

Ingrese el número de iteraciones: -6
Ingrese el coeficiente y el exponente: 2 4
ERROR 3:el número de iteraciones es menor igual a 0

Casos de prueba para verificación de solución

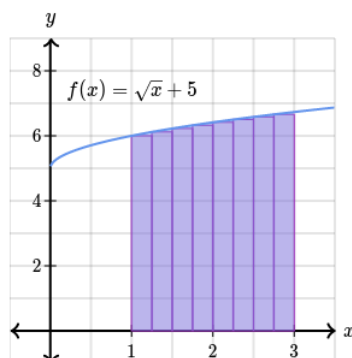
Ingrese el número de iteraciones: 7

Ingrese el coeficiente y el exponente: 1 -5

ERROR 2:el exponente es menor igual a 0

2.3.10. Suma de Riemann (adaptado del laboratorio 5 del ciclo 2020-2)

En la siguiente gráfica se muestra la suma de Riemann y cómo en base a ella es posible hallar el área bajo la curva:



La fórmula general para hallar el área bajo la curva es:

$$\text{area} = \sum_{i=1}^n (i^{\text{exp}} + \text{indep})$$

Se le pide que calcule el área considerando un intervalo $[\text{limiteinferior}, \text{limitesuperior}]$ ingresado por el usuario, e identifique en que opción de valores se encuentra dicha área.

- Las opciones de respuesta son:
- | | |
|----------|---------------------------|
| Opción 1 | $[0, 20[$ |
| Opción 2 | $[20, 70[$ |
| Opción 3 | $[70, 200[$ |
| Opción 4 | $[200, 400[$ |
| Opción 5 | Ninguna de las anteriores |

Se le pide que utilizando programación modular y en al menos cuatro módulos incluido el principal(main) identifique y muestre la opción de respuesta correcta en la que se encuentra el área hallada, como se muestra en el caso de prueba.

Para ello debe leer los límites del intervalo, el exponente y el término independiente. Posteriormente deberá validar los datos (como se especifica líneas más abajo), si estos son válidos deberá calcular el valor de la suma de Riemann. Una vez realizados los cálculos, deberá evaluar cada una de las opciones e ir descartándolas según convenga (ver casos de prueba) y mostrar las opciones descartadas. Finalmente deberá identificar la opción correcta y mostrar el mensaje correspondiente, debe mostrar el área y el detalle de la opción (vea ejemplo de prueba).

Para la validación, debe verificar que todos los valores del intervalo sean válidos y que el exponente sea mayor igual a 0. Si alguno o varios de los datos no son validos deberá mostrar uno de los siguientes mensajes. Recuerde que el máximo valor para una variable de tipo int se ubica en la constante `INT_MAX <limits.h>`. En caso haya más de un error, debe mostrar únicamente el primero ocurrido.

ERROR 1:el exponente es menor a 0

ERROR 2:el límite inferior es menor a 0

ERROR 3:el límite superior es menor a 0

ERROR 4:el límite superior es menor al límite inferior

Casos de prueba para verificación de solución

Ingrese el límite inferior y superior del intervalo: 3
 Ingrese el término independiente y el exponente: 5 0.5
 iteración 1 : suma: 6.000000
 iteración 2 : suma: 12.414214
 iteración 3 : suma: 19.146264
 El área es:19.146264
 opción 1 - Rpta [1,20[

Casos de prueba para verificación de solución

Ingrese el límite inferior y superior del intervalo: 6 -9
 Ingrese el término independiente y el exponente: 5 0.5
 ERROR 3:el límite superior es menor a 0

Casos de prueba para verificación de solución

Ingrese el límite inferior y superior del intervalo: 5 2
 Ingrese el término independiente y el exponente: 6 0.5
 ERROR 4:el límite superior es menor al límite inferior

Casos de prueba para verificación de solución

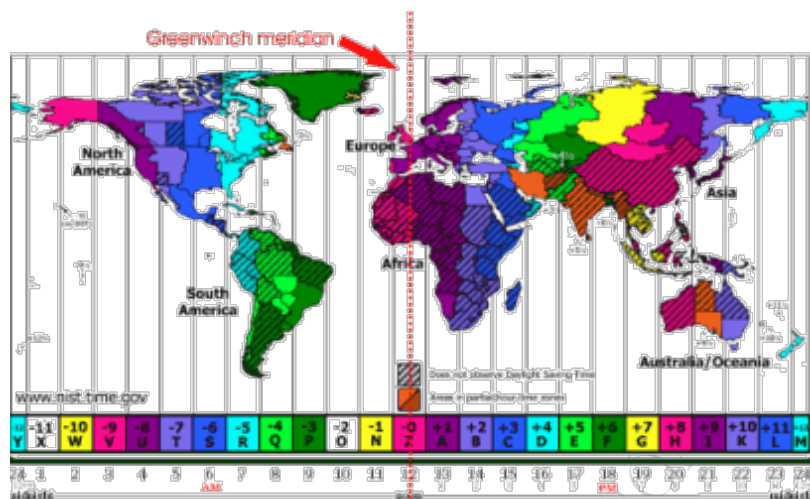
Ingrese el límite inferior y superior del intervalo: -5 6
 Ingrese el término independiente y el exponente: 5 0.5
 ERROR 2:el límite inferior es menor a 0

Casos de prueba para verificación de solución

Use los siguiente casos para verificar si su solución está correcta.
 Ingrese el límite inferior y superior del intervalo: 654 4358
 Ingrese el término independiente y el exponente: 43256 9686
 descartada opción 1,2,3 y 4
 El área es:inf
 Ninguna de las anteriores

2.3.11. Husos horarios versión 1(adaptado del laboratorio 5 del ciclo 2020-2)

Los husos horarios son cada uno de los 24 sectores esféricos en que se divide la superficie de la Tierra. Resultan de repartir los 360° de la esfera terrestre entre las 24 horas que invierte en dar una vuelta completa sobre su propio eje. Cada huso horario mide 15 grados ($360^\circ / 24 \text{ horas} = 15^\circ$). $15^\circ = 1 \text{ hora}$ Para conocer la hora en algún lugar del mundo se toma como referencia el meridiano 0 (Greenwich). A partir de él se añade una hora por cada huso horario que se recorra hacia el este y se resta una hora por cada huso que se recorra hacia el oeste.



Sabemos además que algunos de los países en ciertos husos son:

- UTC – 10. Hawái (W).
- UTC – 8. México (M).
- UTC – 6. Guatemala (G), Nicaragua (N), Honduras(H).
- UTC – 5. Brasil (B), Jamaica (J), Perú (P).
- UTC – 3. Uruguay(U).
- UTC + 1. Alemania(A), Eslovenia(E), Dinamarca(D), Italia(I).
- UTC + 2. Finlandia(F), Zimbabue(Z),Sudáfrica (S).
- UTC + 3. Catar (C), Yemen(Y),Turquía(Q).
- UTC + 4. Omán(O).
- UTC + 5. Kazajistán (K),
- UTC + 7. Laos(L), Rusia (R), Tailandia(T), Vietnam(V)

Fuente: <https://concepto.de/husos-horarios/ixzz6ce9PwUH4>

Al viajar sucede que las horas en diversos países son diferentes, por ejemplo, mientras que en Perú son las 16:31 en España son las 22:31. Esta diferencia horaria se obtiene en base a los husos. Se le pide que utilizando programación modular y en al menos cuatro módulos incluido el principal(main) lea el país de origen del viaje, la hora y los minutos de ese país de origen y el número de escalas que hará para llegar a su destino final. Obtenga la hora y los minutos del dato ingresado por el usuario, verifique que tanto el país de origen, el número de escalas, la hora y los minutos son válidos. Recuerde que un país es válido si el código del país está entre A..Z (vea la lista anterior), la escala debe ser mayor a 0 y la hora y los minutos deben ser menores a 60 y mayores o iguales a 0. Debera mostrar reportes de la validación por medio de mensajes como: el origen es válido

el número de escalas es válido

la hora es válida

Los minutos son válidos

Los minutos son válidos

la hora no es válida

el número de escalas no es válido

el origen no es válido

Si todos los datos ingresados son válidos, deberá hallar el huso al que pertenece el origen, y tantas veces como

escalas existan deberá pedirle al usuario el país de destino para esa escala, hallar el huso de dicho país y calcular la hora del destino, y mostrar la hora y minutos de dicha escala. Para hallar la hora de la escala debe considerar que esa nueva hora es igual a $(abs(husoOrigen - husoDestino) + hora)$

Casos de prueba para verificación de solución

Ingrese el país de origen: P
 Ingrese el número de escalas: 3
 Ingrese la hora y los minutos del país de origen: 1614
 el origen es válido
 el número de escalas es válido
 la hora es válida
 Los minutos son válidos
 Ingrese la escala 1 A
 La hora en la escala es: 2214
 Ingrese la escala 2 L
 La hora en la escala es: 414
 Ingrese la escala 3 Q
 La hora en la escala es: 014

Casos de prueba para verificación de solución

Ingrese el país de origen:
 Ingrese el número de escalas: 4
 Ingrese la hora y los minutos del país de origen: 1704
 el origen no es válido

Casos de prueba para verificación de solución

Ingrese el país de origen: P
 Ingrese el número de escalas: -4
 Ingrese la hora y los minutos del país de origen: 1503
 el origen es válido
 el número de escalas no es válido

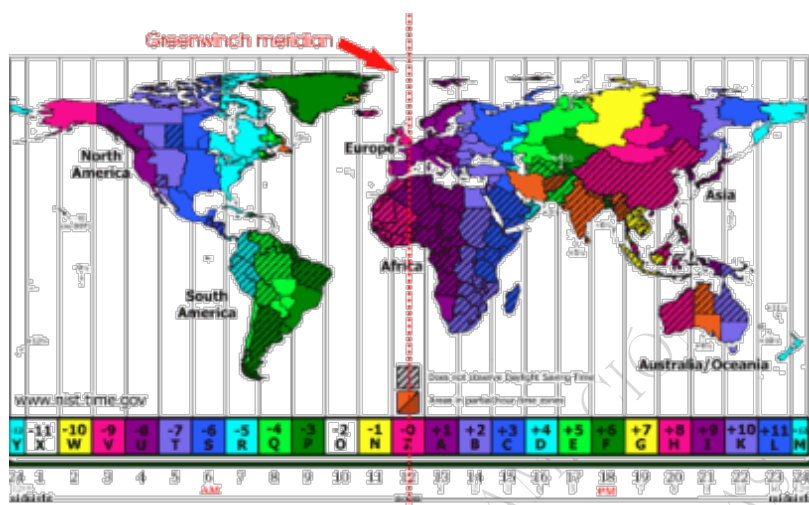
Casos de prueba para verificación de solución

Ingrese el país de origen: P
 Ingrese el número de escalas: 3
 Ingrese la hora y los minutos del país de origen: 7645
 el origen es válido
 el número de escalas es válido
 la hora no es válida

2.3.12. Husos horarios versión 2 (adaptado del laboratorio 5 del ciclo 2020-2)

Los husos horarios son cada uno de los 24 sectores esféricos en que se divide la superficie de la Tierra. Resultan de repartir los 360° de la esfera terrestre entre las 24 horas que invierte en dar una vuelta completa sobre su propio eje. Cada huso horario mide 15 grados ($360^\circ / 24 \text{ horas} = 15^\circ$). $15^\circ = 1 \text{ hora}$ Para conocer la hora en

algún lugar del mundo se toma como referencia el meridiano 0 (Greenwich). A partir de él se añade una hora por cada huso horario que se recorra hacia el este y se resta una hora por cada huso que se recorra hacia el oeste.



Fuente: <https://concepto.de/husos-horarios/ixzz6ce9PwUH4>

Al viajar sucede que las monedas en diversos países son diferentes, por ejemplo, mientras que en Perú usamos el sol, en España se usan los euros.

A continuación se detallan algunas monedas de algunos países:

- Quetzal: Guatemala (G), 1 Quetzal = 0.46 soles
- Euro: Holanda(H), Alemania(A), Eslovenia(E), Lituania (L), Dinamarca(D), Italia(I), Finlandia(F), 1 Euro = 4.21 soles
- Real brasileiro: Brasil (B), 1 real = 0.63 soles
- Dolar jamaquino: Jamaica (J), 1 dolar jamaquino = 0.025 soles
- Sol: Perú (P) 1 sol=1 sol
- Rial qatarí: Catar (C), 1 rial qatarí = 0.99 soles
- Chelin keniano: Kenia (K), 1 chelin keniano = 0.033 soles

Se le pide que utilizando programación modular y en al menos cuatro módulos incluido el principal(main) lea el país de origen del viaje, un monto en la moneda de dicho país y el número de escalas que hará para llegar a su destino final. Debe validar los datos ingresado, el país debe estar entre los países listados (A..L o P), el monto debe ser mayor a 0 y las escalas también. Deberá mostrar reportes de la validación por medio de mensajes como: el origen es válido
el número de escalas es válido
el monto es válido
el monto no es válido
el número de escalas no es válido
el origen no es válido

Si todos los datos ingresados son válidos, deberá hallar el tipo de cambio del origen y convertirlo a soles. Posteriormente, para cada escala deberá pedirle al usuario el país de la escala y el monto gastado en dicho país (en la moneda del país), validar que estos sean válidos, sino lo son, debe detener la ejecución, de ser válidos debe hallar el tipo de cambio para ese país de escala, convertir el monto a soles y mostrar el monto gastado en soles. Si trata de gastar más de lo que tiene, deberá mostrar el mensaje **ERROR**, está tratando de gastar más de lo que tiene, caso contrario deberá calcular y mostrar el monto en la moneda del país de destino. Recuerde que para este laboratorio no puede utilizar la instrucción break, pero para terminar la ejecución de la iterativa puede asignarle a la variable de control el valor maximo, en este caso el número de escalas.

Casos de prueba para verificación de solución

Ingrese el país de origen: W
Ingrese el número de escalas: 4
Ingrese el monto: 1000
el origen no es válido

Casos de prueba para verificación de solución

Ingrese el país de origen: H
Ingrese el número de escalas: -3
Ingrese el monto: 1000
el origen es válido
el número de escalas no es válido

Casos de prueba para verificación de solución

Ingrese el país de origen: H
Ingrese el número de escalas: 4
Ingrese el monto: -100
el origen es válido
el número de escalas es válido
el monto no es válido

Casos de prueba para verificación de solución

Ingrese el país de origen: B
Ingrese el número de escalas: 3
Ingrese el monto: 300
el origen es válido
el número de escalas es válido
el monto es válido
Se cuenta con: 189.00 soles
Ingrese la escala 1 G
Ingrese el monto gastado 1 10
El monto gastado es: 4.60 soles
El nuevo monto en la moneda de destino es: 400.87
Ingrese la escala 2 C
Ingrese el monto gastado 2 100
El monto gastado es: 99.00 soles
El nuevo monto en la moneda de destino es: 86.26
Ingrese la escala 3 P
Ingrese el monto gastado 3 10
El monto gastado es: 10.00 soles
El nuevo monto en la moneda de destino es: 75.40

Casos de prueba para verificación de solución

Ingrese el país de origen: H
 Ingrese el número de escalas: 3
 Ingrese el monto: 1000
 el origen es válido
 el número de escalas es válido
 el monto es válido
 Se cuenta con: 4210.00 soles
 Ingrese la escala 1 W
 Ingrese el monto gastado 1 10
 el origen no es válido

Casos de prueba para verificación de solución

Ingrese el país de origen: H
 Ingrese el número de escalas: 3
 Ingrese el monto: 1000
 el origen es válido
 el número de escalas es válido
 el monto es válido
 Se cuenta con: 4210.00 soles
 Ingrese la escala 1 I
 Ingrese el monto gastado 1 -10
 el origen es válido
 el número de escalas es válido
 el monto no es válido

Casos de prueba para verificación de solución

Ingrese el país de origen: H
 Ingrese el número de escalas: 3
 Ingrese el monto: 100
 el origen es válido
 el número de escalas es válido
 el monto es válido
 Se cuenta con: 421.00 soles
 Ingrese la escala 1 P
 Ingrese el monto gastado 1 500
 el origen es válido
 el número de escalas es válido
 el monto es válido
 El monto gastado es: 500.00 soles
 ERROR, está tratando de gastar más de lo que tiene

2.3.13. Números centralizados (adaptado del laboratorio 5 del ciclo 2020-2)

Los números centralizados son números figurativos que se definen como:

Triangulares centralizados:

$$T_n = \frac{3n(n-1)}{2} + 1$$

Pentagonales centralizados:

$$P_n = \frac{(5n^2 - 5n + 2)}{2}$$

Heptagonales centralizados:

$$H_n = \frac{7n(n-1)}{2} + 1$$

Decagonales centralizados:

$$D_n = 5n(n-1) + 1$$

Se le pide que utilizando programación modular y en al menos cuatro módulos incluido el principal(main) elabore un programa en lenguaje C que permita verificar o calcular el número centralizado para algunas de las siguientes opciones: Triángulos(T), Pentagonos(P), Heptagonos(H), Decagonos(D). Para ello, deberá leer la opción ingresada por el usuario, validarla, de no ser una opción válida, deberá mostrar un mensaje de error Ingreso una opción inválida y terminar la ejecución. Si la opción ingresada es válida deberá reportar la opción seleccionada por medio de mensajes descriptivos:

Ha seleccionado triangulares centralizados

Ha seleccionado pentagonales centralizados

Ha seleccionado heptagonales centralizados

Ha seleccionado decagonales centralizados

Luego deberá leer la acción que realizará el usuario, las acciones pueden ser dos: verificar(v) o calcular(c), valide la acción ingresada por el usuario. Si el usuario selecciona calcular deberá solicitar al usuario un valor para n, validar que este sea mayor a 0 para finalmente calcular y mostrar el número centralizado correspondiente. Por ejemplo, si el usuario ingresó P c 5, se deberá mostrar el mensaje: El P_5 es 51. Si selecciona la acción de verificar deberá solicitar al usuario el valor de n y el supuesto valor del número centralizado y verificar si el supuesto valor es realmente el número centralizado para ese n. Por ejemplo, si el usuario ingresó T v 3 50, se deberá mostrar el mensaje Los valores no coinciden, 10 es el número T_3

Casos de prueba para verificación de solución

Opción:

T:triangulares

P:pentagonales

H:heptagonales

D:decagonal

Ingrese la opción:P

Ha seleccionado pentagonales centralizados

Seleccione la acción a realizar:

v:Verificar

c:Calcular

c

Ingrese el valor de n: 5

El P_5 es 51

Casos de prueba para verificación de solución

Opción:

T:triangulares

P:pentagonales

H:heptagonales

D:decagonal

Ingrese la opción:T

Ha seleccionado triangulares centralizados

Seleccione la acción a realizar:

v:Verificar

c: Calcular

v

Ingrese el valor de n y el supuesto valor del número centralizado: 3 50

Los valores no coinciden, 10 es el número T3

Casos de prueba para verificación de solución

Opción:

T:triangulares

P:pentagonales

H:heptagonales

D:decagonal

Ingrese la opción:T

Ha seleccionado triangulares centralizados

Seleccione la acción a realizar:

v:Verificar

c: Calcular

V

Acción inválida

Casos de prueba para verificación de solución

Opción:

T:triangulares

P:pentagonales

H:heptagonales

D:decagonal

Ingrese la opción:W

Ingreso una opción inválida

2.3.14. Identificar patrón (adaptado del laboratorio 5 del ciclo 2020-2)

Se desea identificar cuántas veces se ubica cierto patrón que representa a un caracter dentro de un número. Y además contabilizar la cantidad de patrones que se encuentran en el.

Para ello se han creado una serie de bloques para ubicar el patrón a modo de caracter:

- Bloque 1: [A..J]
- Bloque 2: [J..L]

- Bloque 3: [L..R[
- Bloque 4: [R..Z]

Se le pide que utilizando programación modular y en al menos cuatro módulos incluido el principal(main) encuentre las veces que aparece el patrón en el número y cuántas veces se encuentran caracteres en cada bloque.

Así en el número 7653659 tenemos el patrón 65 (A), dos veces, ya que analizamos 76, 65, 53, 36, 65, 59. Además, considerando los bloques, se tienen 2 valores para el bloque 1, 0 para el bloque 2, 1 para el bloque 3 y 0 para el bloque 4.

Para ello debe leer el número a evaluar, la cantidad de dígitos que este posee, y el patrón a buscar (siempre formado por dos dígitos), en forma de un caracter [A..Z]. Posteriormente deberá validar los datos (como se especifica líneas más abajo), si estos son válidos deberá calcular la cantidad de apariciones del patrón en el número y la cantidad de veces que se encuentran valores en cada bloque (vea casos de prueba). Luego deberá mostrar que bloque es el que tiene menor cantidad, especificando el bloque y el rango de valores (vea casos de prueba). Para la validación, debe verificar que el número este entre 0 y el valor máximo posible del tipo de dato, además debe verificar que el caracter (patrón) sea uno en mayúsculas entre la A y la Z y mostrar los mensajes éxito o error según convenga para la validación y como se muestra en los casos de prueba. Recuerde que el máximo valor para una variable de tipo int se ubica en la constante INT_MAX <limits.h>.

Los mensajes a considerar son:

El número es mayor a 0

El número es menor al límite del tipo de dato

El caracter ingresado es válido

El caracter ingresado no es válido

El número es mayor o igual al límite del tipo de dato

El número es menor a 0 (pudo deberse al desbordamiento de dato)

Casos de prueba para verificación de solución

Ingrese el número, la cantidad de dígitos, el patrón a buscar (un caracter): 65378726 8 A

El número es mayor a 0

El número es menor al límite del tipo de dato

El caracter ingresado es válido

El caracter ingresado se encontró 1 veces en el número

El bloque 1 se encuentra 2 veces

El bloque 2 se encuentra 0 veces

El bloque 3 se encuentra 1 veces

El bloque 4 se encuentra 1 veces

El bloque 2 [J,L[presenta el menor valor de todos

Casos de prueba para verificación de solución

Ingrese el número, la cantidad de dígitos, el patrón a buscar (un caracter): 67837843278423 14 G

El número es menor a 0 (pudo deberse al desbordamiento de dato)

Casos de prueba para verificación de solución

Ingrese el número, la cantidad de dígitos, el patrón a buscar (un caracter): 76345 5 9
El número es mayor a 0
El número es menor al límite del tipo de dato
El caracter ingresado no es válido

Casos de prueba para verificación de solución

Ingrese el número, la cantidad de dígitos, el patrón a buscar (un caracter): -54343 5 U
El número es menor a 0 (pudo deberse al desbordamiento de dato)

Casos de prueba para verificación de solución

Ingrese el número, la cantidad de dígitos, el patrón a buscar (un caracter): 2147483647 9 H
El número es mayor a 0
El número es mayor o igual al límite del tipo de dato

Casos de prueba para verificación de solución

Ingrese el número, la cantidad de dígitos, el patrón a buscar (un caracter): 7653659 7 A
El número es mayor a 0
El número es menor al límite del tipo de dato
El caracter ingresado es válido
El caracter ingresado se encontró 2 veces en el número
El bloque 1 se encuentra 2 veces
El bloque 2 se encuentra 0 veces
El bloque 3 se encuentra 1 veces
El bloque 4 se encuentra 0 veces
El bloque 2 [J,L] presenta el menor valor de todos El bloque 4 [R,Z] presenta el menor valor de todos

Casos de prueba para verificación de solución

Ingrese el número, la cantidad de dígitos, el patrón a buscar (un caracter): 654356 6 B
El número es mayor a 0
El número es menor al limite del tipo de dato
El caracter ingresado es válido
El caracter ingresado se encontró 0 veces en el número
El bloque 1 se encuentra 1 veces
El bloque 2 se encuentra 0 veces
El bloque 3 se encuentra 0 veces
El bloque 4 se encuentra 0 veces
El bloque 2 [J,L] presenta el menor valor de todos

2.3.15. Encriptar o desencriptar un mensaje escondido (adaptado del laboratorio 5 del ciclo 2020-2)

Se desea encriptar o desencriptar un mensaje escondido en un número. Esto se logra haciendo uso del código ASCII de cada caracter. Para ello, se le pide que utilizando programación modular y en al menos cuatro

módulos incluido el principal(main) lea el número a evaluar, la cantidad de dígitos de dicho número, la opción (1: encriptar o 2:desencriptar) y el código a utilizar (este código expresa la cantidad de caracteres utilizados en el desplazamiento),posteriormente debe validar estos datos, encriptar o desencriptar dependiendo de la opción del usuario y mostrar las salidas como se muestra en los casos de prueba.

Debe considerar que para la validación de los datos de entrada, el número debe ser mayor a 0 y menor al valor máximo del tipo de dato, recuerde que el máximo valor para una variable de tipo int se ubica en la constante INT_MAX <limits.h>. La opción debe ser o 1 para encriptar o 2 para desencriptar, el código para encriptar o desencriptar debe ser un número mayor a 0 y menor a 26. La cantidad de dígitos debe ser un número par. Este código corresponde al número de caracteres que debe desplazarse. El análisis se realiza de dos en dos dígitos empezando por la izquierda. Por ejemplo, el mensaje 65666768, con opción 1 y código 2 es: 65 que corresponde al caracter A se vuelve 67 (considerando un desplazamiento de 2) que corresponde al caracter C, 66 que corresponde al caracter B se vuelve 68 que corresponde al caracter D, 67 que corresponde al caracter C se vuelve 69 que corresponde al caracter E, y 68 que corresponde al caracter D se vuelve 70 que corresponde al caracter F. Si luego del desplazamiento se excede al alfabeto debe volver al inicio o al fin de dicho alfabeto, según corresponda, por ejemplo W con código 4 de encriptación debe dar como resultado A y en el caso de A con código 4 de desencriptación debe dar como resultado W.

Si los datos de entrada son inválidos debe mostrar el mensaje de error correspondiente:

ERROR todos los datos de ingreso están equivocados

ERROR fuera de rango y opción inválida

ERROR fuera de rango y código inválido

ERROR opción inválida y código inválido

ERROR solo un dato de entrada es incorrecto

Si el caracter no está en el alfabeto [A..Z] debe mostrar el mensaje correspondiente: No está en el alfabeto (Ver casos de prueba)

Casos de prueba para verificación de solución

Opciones:

(1)Encriptar

(2)Desencriptar

Ingrese el número, la cantidad de dígitos, la opción y el código:67876569 8 1 3

E 69 H 72

A 65 D 68

W 87 Z 90

C 67 F 70

Evaluando encriptado:

Código 3:

Mensaje original: 67876569

Mensaje encriptado: 69658767

Casos de prueba para verificación de solución

Opciones:

(1)Encriptar

(2)Desencriptar

Ingrese el número, la cantidad de dígitos, la opción y el código:92748569 8 2 4

E 69 A 65

U 85 Q 81

J 74 F 70

92 No está en el alfabeto

Evaluando desencriptado:

Código 4:

Mensaje encriptado: 92748569

Mensaje original: 65817099

Casos de prueba para verificación de solución

Opciones:

(1)Encriptar

(2)Desencriptar

Ingrese el número, la cantidad de dígitos, la opción y el código:-546783 6 1 3

ERROR solo un dato de entrada es incorrecto

Casos de prueba para verificación de solución

Opciones:

(1)Encriptar

(2)Desencriptar

Ingrese el número, la cantidad de dígitos, la opción y el código:6746553 7 3 6

ERROR solo un dato de entrada es incorrecto

2.3.16. Identificar la cantidad de dígitos (adaptado del laboratorio 5 del ciclo 2020-2)

Se desea identificar la cantidad de dígitos pares o impares de un número y evaluar si dicho número es divisible entre cierto dígito.

Para ello se han creado una serie de opciones que se mostrarán al usuario para saber que se desea evaluar:

- Evaluar solo par (1).
- Evaluar solo impar (2).
- Evaluar par y divisible (3).
- Evaluar impar y divisible (4).

Se le pide que utilizando programación modular y en al menos cuatro módulos incluido el principal(main) identifique y muestre el resultado para la opción solicitada por el usuario, como se muestra en el caso de prueba.

Para ello debe leer el número a evaluar, la cantidad de dígitos que posee dicho número, la opción y el dígito divisor y reportar la opción que está evaluando. Posteriormente deberá validar los datos (como se especifica líneas más abajo), si estos son válidos deberá calcular la cantidad de dígitos pares e impares del número y dependiendo del valor de la opción mostrar el mensaje adecuado (Ver casos de prueba).

Para la validación, debe verificar que el número este entre 0 y el valor máximo posible del tipo de dato, que la opción sea una de las opciones posibles y el dígito divisor sea mayor a 0 y menor a 10 (sin importar la opción

ingresada por el usuario). Recuerde que el máximo valor para una variable de tipo int se ubica en la constante INT_MAX <limits.h>.

En caso de existir errores debe reportarlos de la siguiente manera:

ERROR se encontró un error en los datos de ingreso

ERROR se encontraron dos errores en los datos de ingreso

ERROR todos los datos de ingreso están equivocados

Casos de prueba para verificación de solución

Opciones:

Solo par (1)

Solo impar (2)

Par y divisible (3)

Impar y divisible (4)

Ingresa el número, la cantidad de dígitos, la opción y el divisor: 237884532 9 4 2

Evaluando Impar y Divisor

El número tiene 4 dígitos impares

El número 237884532 es divisible entre 2

Casos de prueba para verificación de solución

Opciones:

Solo par (1)

Solo impar (2)

Par y divisible (3)

Impar y divisible (4)

Ingresa el número, la cantidad de dígitos, la opción y el divisor: 832945 6 4 3

Evaluando Impar y Divisor

El número tiene 3 dígitos impares

El número 832945 no es divisible entre 3

Casos de prueba para verificación de solución

Opciones:

Solo par (1)

Solo impar (2)

Par y divisible (3)

Impar y divisible (4)

Ingresa el número, la cantidad de dígitos, la opción y el divisor: 23456443 8 1 7

Evaluando Solo par

El número tiene 5 dígitos pares

No se uso el divisor 7

Casos de prueba para verificación de solución

Opciones:
Solo par (1)
Solo impar (2)
Par y divisible (3)
Impar y divisible (4)
Ingrese el número, la cantidad de dígitos, la opción y el divisor: 78943763 8 2 4
Evaluando Solo impar
El número tiene 5 dígitos impares
No se uso el divisor 4

Casos de prueba para verificación de solución

Opciones:
Solo par (1)
Solo impar (2)
Par y divisible (3)
Impar y divisible (4)
Ingrese el número, la cantidad de dígitos, la opción y el divisor: 31987495 8 3 9
Evaluando Par y Divisor
El número tiene 2 dígitos pares
El número 31987495 no es divisible entre 9

Casos de prueba para verificación de solución

Opciones:
Solo par (1)
Solo impar (2)
Par y divisible (3)
Impar y divisible (4)
Ingrese el número, la cantidad de dígitos, la opción y el divisor: -678433 6 8 -2
ERROR todos los datos de ingreso están equivocados

Casos de prueba para verificación de solución

Use los siguiente casos para verificar si su solución está correcta.
Ingrese el número, la cantidad de dígitos, la opción y el divisor: 576764 6 8 0
ERROR se encontraron dos errores en los datos de ingreso

Casos de prueba para verificación de solución

Opciones:
Solo par (1)
Solo impar (2)
Par y divisible (3)
Impar y divisible (4)
Ingrese el número, la cantidad de dígitos, la opción y el divisor: 987643222423445578 18 6 4
ERROR se encontró un error en los datos de ingreso

2.3.17. Condensador (adaptado del laboratorio 5 del ciclo 2020-2)

Un capacitor o también conocido como condensador es un dispositivo capaz de almacenar energía a través de campos eléctricos (uno positivo y uno negativo). Su valor se expresa en faradios en honor al físico británico Michael Faraday. En la figura 2.2 se puede apreciar a una serie de capacitores. Como podrá notar, en algunos capacitores solamente aparece un número colocado en el cuerpo del mismo, este número representa al valor del capacitor siguiendo determinada codificación.

Los capacitores se pueden colocar tanto en serie como en paralelo, en cuyo caso la capacitancia total se debe calcular usando unas fórmulas. Cuando las capacitores se colocan en serie, el valor total de la capacitancia C_T se calcula de la siguiente manera:

$$C_T = \frac{1}{\sum_{i=1}^n \frac{1}{C_i}}$$

Donde C_i es la capacitancia cada capacitor que compone el sistema que se desea analizar.

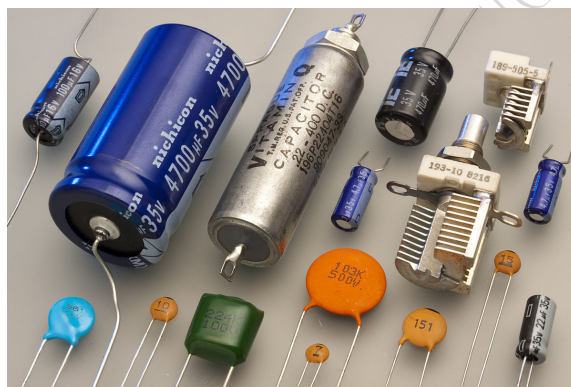


Figura 2.2: Capacitores. Imagen obtenida del URL https://es.wikipedia.org/wiki/Condensador_el%C3%A9ctrico.

Una empresa distribuidora de capacitores ha decidido incrementar su inventario, pero para ello necesita saber con que capacitores cuenta organizándolos por rangos en picofaradios.

Rango 1: Menor a 1000

Rango 2: Mayor igual a 1000 y menor a 2000

Rango 3: Mayor igual a 2000 y menor a 5000

Rango 4: Mayor igual a 5000 y menor a 14000

Rango 5: Mayor igual a 14000 y menor a más

Se sabe que un capacitor posee un código con la información del capacitor. El código de un capacitor es muy simple de entender. De las 3 cifras que representan al código, la cifra de las unidades corresponde al exponente que se debe elevar el número 10. Luego deberá multiplicar por el número que se obtiene de eliminar la cifras de las unidades por esta potencia de 10 para obtener el valor del capacitor en picofaradios. Por ejemplo, si el código de un capacitor fuera 472, la cifras de las unidades es 2, el número que se obtiene de eliminar las unidades es 47, entonces el valor en picofaradios del capacitor sería $47 \times 10^2 = 4700$.

Se le pide que utilizando programación modular y en al menos cuatro módulos incluido el principal(main) lea la cantidad de capacitores a registrar, verifique que esta cantidad sea mayor a 0, si no es mayor a 0 debe mostrar un mensaje de error. Si es mayor a 0, entonces deberá proceder a solicitar al usuario el código de cada uno de los capacitores, verificar si el código es de tres cifras, si no lo es mostrar un mensaje de error. De ser válido el código, debe proceder a calcular el valor de la capacitancia en picofaradios para el mismo y actualizar el inventario en el rango adecuado y mostrar la cantidad de capacitores para cada rango. Finalmente deberá validar que en cada rango haya al menos el 25 por ciento de la cantidad de capacitores registrados, de no haberlos deberá mostrar los siguientes mensajes con indicaciones para las compras que se deben hacer para asegurar esta mínima cantidad. Todos los rangos de capacitores necesitan adquirir más capacitores. Ningún rango de capacitores necesita adquirir más capacitores.

Solo para el rango X se necesita comprar capacitores donde X es el identificador del rango (ver caso de prueba)

Tres de los rangos de capacitores necesita adquirir más capacitores

Dos de los rangos de capacitores necesita adquirir más capacitores

Recuerde que para este laboratorio no puede utilizar la instrucción break, pero para terminar la ejecución de la iterativa puede asignarle a la variable de control el valor maximo, en este caso el número de capacitores.

Casos de prueba para verificación de solución

Ingrese la cantidad de capacitores: 4
Ingrese código del capacitor: 120
El valor de la capacitancia es 12.0000 pF
Rango1=1 Rango2=0 Rango3=0 Rango4=0 Rango5=0
Ingrese código del capacitor: 120
El valor de la capacitancia es 12.0000 pF
Rango1=2 Rango2=0 Rango3=0 Rango4=0 Rango5=0
Ingrese código del capacitor: 544
El valor de la capacitancia es 540000.0000 pF
Rango1=2 Rango2=0 Rango3=0 Rango4=0 Rango5=1
Ingrese código del capacitor: 544
El valor de la capacitancia es 540000.0000 pF
Rango1=2 Rango2=0 Rango3=0 Rango4=0 Rango5=2
Tres de los rangos de capacitores necesita adquirir más capacitores

Casos de prueba para verificación de solución

Ingrese la cantidad de capacitores: 4
Ingrese código del capacitor: 4255
Código del capacitor inválido, vuelva a iniciar el registro

Casos de prueba para verificación de solución

Ingrese la cantidad de capacitores: -49
la cantidad debe ser mayor a 0

Casos de prueba para verificación de solución

Ingrese la cantidad de capacitores: 10
Ingrese código del capacitor: 431
El valor de la capacitancia es 430.0000 pF
Rango1=1 Rango2=0 Rango3=0 Rango4=0 Rango5=0
Ingrese código del capacitor: 876
El valor de la capacitancia es 87000000.0000 pF
Rango1=1 Rango2=0 Rango3=0 Rango4=0 Rango5=1
Ingrese código del capacitor: 541
El valor de la capacitancia es 540.0000 pF
Rango1=2 Rango2=0 Rango3=0 Rango4=0 Rango5=1
Ingrese código del capacitor: 543
El valor de la capacitancia es 54000.0000 pF
Rango1=2 Rango2=0 Rango3=0 Rango4=0 Rango5=2
Ingrese código del capacitor: 182
El valor de la capacitancia es 1800.0000 pF
Rango1=2 Rango2=1 Rango3=0 Rango4=0 Rango5=2
Ingrese código del capacitor: 871
El valor de la capacitancia es 870.0000 pF
Rango1=3 Rango2=1 Rango3=0 Rango4=0 Rango5=2
Ingrese código del capacitor: 112
El valor de la capacitancia es 1100.0000 pF
Rango1=3 Rango2=2 Rango3=0 Rango4=0 Rango5=2
Ingrese código del capacitor: 152
El valor de la capacitancia es 1500.0000 pF
Rango1=3 Rango2=3 Rango3=0 Rango4=0 Rango5=2
Ingrese código del capacitor: 252
El valor de la capacitancia es 2500.0000 pF
Rango1=3 Rango2=3 Rango3=1 Rango4=0 Rango5=2
Ingrese código del capacitor: 434
El valor de la capacitancia es 430000.0000 pF
Rango1=3 Rango2=3 Rango3=1 Rango4=0 Rango5=3
Dos de los rangos de capacitores necesita adquirir más capacitores

Casos de prueba para verificación de solución

Ingrese la cantidad de capacitores: 8
 Ingrese código del capacitor: 152
 El valor de la capacitancia es 1500.0000 pF
 Rango1=0 Rango2=1 Rango3=0 Rango4=0 Rango5=0
 Ingrese código del capacitor: 152
 El valor de la capacitancia es 1500.0000 pF
 Rango1=0 Rango2=2 Rango3=0 Rango4=0 Rango5=0
 Ingrese código del capacitor: 320
 El valor de la capacitancia es 32.0000 pF
 Rango1=1 Rango2=2 Rango3=0 Rango4=0 Rango5=0
 Ingrese código del capacitor: 320
 El valor de la capacitancia es 32.0000 pF
 Rango1=2 Rango2=2 Rango3=0 Rango4=0 Rango5=0
 Ingrese código del capacitor: 502
 El valor de la capacitancia es 5000.0000 pF
 Rango1=2 Rango2=2 Rango3=0 Rango4=1 Rango5=0
 Ingrese código del capacitor: 502
 El valor de la capacitancia es 5000.0000 pF
 Rango1=2 Rango2=2 Rango3=0 Rango4=2 Rango5=0
 Ingrese código del capacitor: 653
 El valor de la capacitancia es 65000.0000 pF
 Rango1=2 Rango2=2 Rango3=0 Rango4=2 Rango5=1
 Ingrese código del capacitor: 653
 El valor de la capacitancia es 65000.0000 pF
 Rango1=2 Rango2=2 Rango3=0 Rango4=2 Rango5=2
 Solo para el rango3 se necesita comprar capacitores

2.3.18. La resistencia (adaptado del laboratorio 5 del ciclo 2020-2)

La resistencia es un dispositivo eléctrico que tiene la particularidad de oponerse al flujo de la corriente. Su valor se expresa en ohmios en honor al físico alemán Georg Simon Ohm. En la figura 2.3 se puede apreciar un conjunto de resistencias. Como podrá notar, existen bandas de colores en el cuerpo de cada una de ellas. Estas bandas de colores permiten determinar el valor que posee la resistencia. De forma general se puede decir que cada color representa al valor de un dígito.

Las resistencias se pueden colocar tanto en serie como en paralelo, en cuyo caso la resistencia total se debe calcular usando unas fórmulas. Cuando las resistencias se colocan en serie, el valor total de la resistencia R_T se calcula de la siguiente manera:

$$R_T = \sum_{i=1}^n R_i$$

Donde R_i es cada resistencia que compone el sistema que se desea analizar.

Una empresa distribuidora de resistencias ha decidido incrementar su inventario, pero para ello necesita saber con que resistencias cuenta organizándolos por rangos en ohmios.

Se le pide que utilizando programación modular y en al menos cuatro módulos incluido el principal(main) lea la cantidad de resistencias a registrar, verifique que esta cantidad sea mayor a 0, si no es mayor a 0 debe mostrar un mensaje de error. Si es mayor a 0, entonces deberá proceder a solicitar al usuario color de la banda 1 y la 2 y el multiplicador. Debe verificar que los datos ingresados sean válidos, de no serlo debe mostrar mensajes explicando el error en particular. Si los datos son válidos, debe calcular la resistencia en Ohmios y registrar la resistencia en el rango adecuado, posteriormente, debe mostrar un reporte con todas las cantidades en todos los rangos y repetir esta acción en cada resistencia. Finalmente debe reportar las resistencias de que rango deben ser adquiridas, considerando que se debe contar en el inventario con al menos 30 por ciento de la cantidad de las resistencias ingresadas para cada rango.

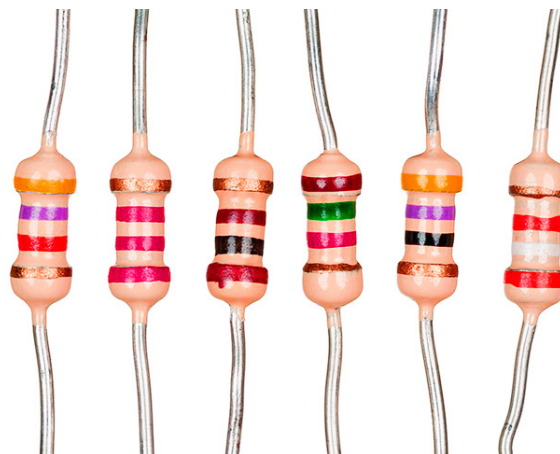


Figura 2.3: Conjunto de resistencias. Imagen obtenida del URL <https://sooluciona.com/diferencias-entre-condensador-y-resistencia/>.

Rango 1: Menor a 100

Rango 2: Mayor igual a 100 y menor a 1000

Rango 3: Mayor igual a 1000 y menor a 4000

Rango 4: Mayor igual a 4000 y menor a 10000

Rango 5: Mayor igual a 10000 y menor a más

Para calcular el valor de la resistencia en ohmios, debe considerar que cada color se representará por un carácter y el multiplicador será un número entero múltiplo de 10 o el 1.

Para el caso de los colores, los colores válidos son: Marrón, representado por el caracter 'B', Rojo, representado por el caracter 'R', Naranja, representado por el caracter 'O' y Amarillo representado por el caracter 'Y'. El color Marrón representa el valor 1, el color Rojo representa el valor 2, el color Naranja representa el valor 3 y el color Amarillo representa el valor 4. Para el caso del multiplicador, deberá validar que este sea 1 o un número positivo múltiplo de 10. En caso no se cumpla con esta validación, deberá emitirse el mensaje El multiplicador ingresado no es válido. Si el multiplicador o el color de la banda no es válido, deberá terminar la ejecución. Para calcular el valor de cada resistencia, primero deberá formar un número usando las bandas de colores: el color de la banda 1 representará el dígito de las decenas y el color de la banda 2 representará el dígito de las unidades. Luego, el número formado deberá multiplicarlo por el multiplicador. El resultado será el valor de la resistencia.

Recuerde que para este laboratorio no puede utilizar la instrucción break, pero para terminar la ejecución de la iterativa puede asignarle a la variable de control el valor maximo, en este caso el cantidad de resistencias.

Casos de prueba para verificación de solución

Ingrese la cantidad de resistencias: -39

La cantidad de resistencias debe ser mayor que uno y menor que o igual a 10

Casos de prueba para verificación de solución

Ingrese la cantidad de resistencias: 4

Datos de la resistencia 1

Ingrese color de la banda 1 y 2 y multiplicador: P Y 10

ERROR en banda 1

Casos de prueba para verificación de solución

Ingrese la cantidad de resistencias: 10
Datos de la resistencia 1
Ingrese color de la banda 1 y 2 y multiplicador: B Y 10
Resistencia: 140
Rango1=0 Rango2=1 Rango3=0 Rango4=0 Rango5=0
Datos de la resistencia 2
Ingrese color de la banda 1 y 2 y multiplicador: O B 1
Resistencia: 31
Rango1=1 Rango2=1 Rango3=0 Rango4=0 Rango5=0
Datos de la resistencia 3
Ingrese color de la banda 1 y 2 y multiplicador: R O 10
Resistencia: 230
Rango1=1 Rango2=2 Rango3=0 Rango4=0 Rango5=0
Datos de la resistencia 4
Ingrese color de la banda 1 y 2 y multiplicador: O Y 100
Resistencia: 3400
Rango1=1 Rango2=2 Rango3=1 Rango4=0 Rango5=0
Datos de la resistencia 5
Ingrese color de la banda 1 y 2 y multiplicador: R Y 1000
Resistencia: 24000
Rango1=1 Rango2=2 Rango3=1 Rango4=0 Rango5=1
Datos de la resistencia 6
Ingrese color de la banda 1 y 2 y multiplicador: R O 100
Resistencia: 2300
Rango1=1 Rango2=2 Rango3=2 Rango4=0 Rango5=1
Datos de la resistencia 7
Ingrese color de la banda 1 y 2 y multiplicador: R O 10
Resistencia: 230
Rango1=1 Rango2=3 Rango3=2 Rango4=0 Rango5=1
Datos de la resistencia 8
Ingrese color de la banda 1 y 2 y multiplicador: Y B 10
Resistencia: 410
Rango1=1 Rango2=4 Rango3=2 Rango4=0 Rango5=1
Datos de la resistencia 9
Ingrese color de la banda 1 y 2 y multiplicador: Y O 10
Resistencia: 430
Rango1=1 Rango2=5 Rango3=2 Rango4=0 Rango5=1
Datos de la resistencia 10
Ingrese color de la banda 1 y 2 y multiplicador: B B 10
Resistencia: 110
Rango1=1 Rango2=6 Rango3=2 Rango4=0 Rango5=1
Dos de los rangos de resistencias necesita adquirir más o resistencias

Casos de prueba para verificación de solución

Ingrese la cantidad de resistencias: 5
Datos de la resistencia 1
Ingrese color de la banda 1 y 2 y multiplicador: Y B 123
El multiplicador ingresado no es válido

Casos de prueba para verificación de solución

Ingrese la cantidad de resistencias: 4
 Datos de la resistencia 1
 Ingrese color de la banda 1 y 2 y multiplicador: Y U 10
 ERROR en banda 2

2.3.19. Colores en HTML versión 1 (adaptado del laboratorio 5 del ciclo 2020-2)

En HTML un color es construido utilizando una combinación de tres colores primarios, rojo (R), verde(G) y azul(B). Cada uno de estos tres colores puede tomar un valor entre 0 y 255 que representa su participación en la mezcla de colores, siendo 255 la participación máxima y 0 la mínima.

Entonces, un código de color utilizado en HTML se construye con un signo numeral("#") seguido de la intensidad de los colores rojo, verde y azul (en ese orden RGB), escritos en notación hexadecimal. Cuando un color en la escala que va desde 0 a 255 se escribe en notación hexadecimal, se obtiene, siempre, un valor de dos cifras entre 00 y FF. HTML considera mayúsculas y minúsculas, pero para este ejercicio debe considerarse solo mayúsculas.

En este esquema, el código de color del rojo más intenso requiere tener el valor "FF" para la parte roja y "00" para las demás, con lo que se obtiene "#FF0000". Asimismo, el código del color verde es "#00FF00" y el del azul "#0000FF". Cuando se mezclan estos tres valores se obtienen resultados equivalentes al de las mezclas en una paleta de pintor, por lo que se puede crear, por ejemplo, amarillo mezclando rojo y verde ("FFFF00"), violeta mezclando rojo y azul ("FF00FF") o cyan mezclando verde y azul ("00FFFF").

El tono del color también puede ser alterado, aumentando o disminuyendo los tres valores proporcionalmente. Por ejemplo, al color violeta ("CC00CC") se puede aclarar ("FF00FF") u oscurecer ("AA00AA").

Se le pide que utilizando programación modular y en al menos cuatro módulos incluido el principal(main) lea una opción : 1) aclarar 2) oscurecer (considere que el aclarar implica un aumento y el oscurecer un decremento del código, además debe leer el nivel de tonalidad para aclarar u oscurecer, debe validar la opción ingresada, sino es válida deberá mostrar el mensaje **Opción inválida**. El tono ingresado debe ser mayor a 0, de no serlo deberá mostrar **tonalidad debe ser mayor a 0**. Si la opción y la tonalidad son válidas, deberá para cada color (R, G y B), solicitar el código, validarlo, si no es válido deberá reportar el detalle del código **Primer código R es inválido**, mostrar el mensaje de error **ERROR en ingreso de datos** y terminar la ejecución, si es válido deberá convertirlo a número para aplicar la tonalidad y luego volver a convertirlo a hexadecimal para finalmente mostrar el nuevo código resultante de aclarar u oscurecer la tonalidad dada, indicando si se oscureció o se aclaró el tono. Asuma que los códigos ingresados por el usuario siempre podrán ser modificados de tonalidad (así el usuario nunca trataría de aclarar un código FFFFFFFF).

Casos de prueba para verificación de solución

Opciones: 1)Aclarar 2)Oscurecer
 Ingrese la opción, y el nivel de tonalidad a modificar: 2 1
 Ingrese el código de R: XX
 Primer código de R inválido
 ERROR en ingreso de datos!

Casos de prueba para verificación de solución

Opciones: 1)Aclarar 2)Oscurecer
 Ingrese la opción, y el nivel de tonalidad a modificar: 3 1
 Opción inválida

Casos de prueba para verificación de solución

Opciones: 1)Aclarar 2)Oscurecer
Ingrese la opción, y el nivel de tonalidad a modificar: 1 3
Ingrese el código de R: CC
Ingrese el código de G: 00
Ingrese el código de B: DD
El color aclarado es:CF03E0

Casos de prueba para verificación de solución

Opciones: 1)Aclarar 2)Oscurecer
Ingrese la opción, y el nivel de tonalidad a modificar: 1 3
Ingrese el código de R: CC
Ingrese el código de G: CC
Ingrese el código de B: CC
El color aclarado es:CFCFCF

Casos de prueba para verificación de solución

Opciones: 1)Aclarar 2)Oscurecer
Ingrese la opción, y el nivel de tonalidad a modificar: 2 2
Ingrese el código de R: FF
Ingrese el código de G: FF
Ingrese el código de B: FF
El color oscurecido es:FDFDFD

2.3.20. Colores en HTML versión 2 (adaptado del laboratorio 5 del ciclo 2020-2)

En HTML un color es construido utilizando una combinación de tres colores primarios, rojo (R), verde(G) y azul(B). Cada uno de estos tres colores puede tomar un valor entre 0 y 255 que representa su participación en la mezcla de colores, siendo 255 la participación máxima y 0 la mínima.

Entonces, un código de color utilizado en HTML se construye con un signo numeral("#") seguido de la intensidad de los colores rojo, verde y azul (en ese orden RGB), escritos en notación hexadecimal. Cuando un color en la escala que va desde 0 a 255 se escribe en notación hexadecimal, se obtiene, siempre, un valor de dos cifras entre 00 y FF. HTML considera mayúsculas y minúsculas, pero para este ejercicio debe considerar solo mayúsculas.

En este esquema, el código de color del rojo más intenso requiere tener el valor "FF" para la parte roja y "00" para las demás, con lo que se obtiene "#FF0000". Asimismo, el código del color verde es ("#00FF00") y el del azul "#0000FF". Cuando se mezclan estos tres valores se obtienen resultados equivalentes al de las mezclas en una paleta de pintor, por lo que se puede crear, por ejemplo, amarillo mezclando rojo y verde ("FFFF00"), violeta mezclando rojo y azul ("FF00FF") o cyan mezclando verde y azul ("00FFFF").

Se le pide que utilizando programación modular y en al menos cuatro módulos incluido el principal(main) lea el código hexadecimal de cada color rojo (R), verde(G) y azul(B), , formado por dos caracteres, valide que los códigos hexadecimales ingresados sean válidos.

De no serlo, deberá mostrar el mensaje según corresponda :

- Si los tres colores con valores inválidos ERROR RGB ingresados están equivocados.
- Si dos de los colores son inválidos: ERROR RB ingresados inválidos, ERROR GB ingresados inválidos, ERROR RG ingresados inválidos.

- Si un color es inválido ERROR solo R o G o B es incorrecto.

Si los datos son válidos deberá obtener la participación del color (el color en base 10, en un número de 9 dígitos) y mostrarlo. Si alguno de los colores tiene una participación máxima o mínima deberá reportarlo como se muestra en los casos de prueba. Recuerde que debe usar en su solución al menos una estructura iterativa (sin utilizar estructuras iterativas anidadas), considere que siempre recibirá 6 caracteres para describir los colores.

Casos de prueba para verificación de solución

Ingrese los códigos para R:54
Ingrese los códigos para G:##
Ingrese los códigos para B:KP
ERROR participación GB ingresados inválidos

Casos de prueba para verificación de solución

Ingrese los códigos para R:TY
Ingrese los códigos para G:24
Ingrese los códigos para B:ZZ
ERROR participación RB ingresados inválidos

Casos de prueba para verificación de solución

Ingrese los códigos para R:GG
Ingrese los códigos para G:HT
Ingrese los códigos para B:21
ERROR participación RG ingresados inválidos

Casos de prueba para verificación de solución

Ingrese los códigos para R:FF
Ingrese los códigos para G:00
Ingrese los códigos para B:AA
Participación máxima: ROJO
Participación mínima: VERDE
El código RGB es 255000170

2.3.21. Colores en HTML versión 3 (adaptado del laboratorio 5 del ciclo 2020-2)

En HTML un color es construido utilizando una combinación de tres colores primarios, rojo (R), verde (G) y azul (B). Cada uno de estos tres colores puede tomar un valor entre 0 y 255 que representa su participación en la mezcla de colores, siendo 255 la participación máxima y 0 la mínima.

Entonces, un código de color utilizado en HTML se construye con un signo numeral ("#") seguido de la intensidad de los colores rojo, verde y azul (en ese orden RGB), escritos en notación hexadecimal. Cuando un color en la escala que va desde 0 a 255 se escribe en notación hexadecimal, se obtiene, siempre, un valor de dos cifras entre 00 y FF. HTML considera mayúsculas y minúsculas, pero para este ejercicio debe considerar solo mayúsculas.

En este esquema, el código de color del rojo más intenso requiere tener el valor "FF" para la parte roja y "00" para las demás, con lo que se obtiene "#FF0000". Asimismo, el código del color verde es ("#00FF00") y el del azul "#0000FF". Cuando se mezclan estos tres valores se obtienen resultados equivalentes al de las mezclas en una paleta de pintor, por lo que se puede crear, por ejemplo, amarillo mezclando rojo y verde ("FFFF00"), violeta mezclando rojo y azul ("FF00FF") o cyan mezclando verde y azul ("00FFFF").

Se le pide que utilizando programación modular y en al menos cuatro módulos incluido el principal(main) lea el código y la cantidad de dígitos que posee dicho código. El código representa los valores seleccionados de la paleta según su posición.

Paleta de colores (posición del color):

- 1.Rojo
- 2.Verde
- 3.Azul
- 4.Cyan
- 5.Amarillo
- 6.Violeta

(p.e. el código: 10010 representa G A, verde y amarillo, esto considerando la paleta de colores y las posiciones 123456) asuma que solo 2 colores pueden ser marcados en 1 y que el código contendrá solo 0s y 1s. Una vez obtenido el código como número deberá encontrar los caracteres que representan a los dos colores utilizados de la paleta p.e. G y A. Recuerde utilizar estructuras algorítmicas iterativas para recorrer el código. Posteriormente, deberá mostrar los caracteres de los colores identificados y verificar si son o no colores primarios y mostrar mensajes descriptivos: El color B es color primario Azul, El color R es color primario Rojo, El color G es color primario Verde.

Si no es un color primario deberá mostrar el mensaje El color X es obtenido por la mezcla de dos colores primarios, donde X debe ser reemplazado por el color identificado (A o V o C).

Finalmente, si ambos colores ingresados son primarios deberá reportar que color se crea, Con color1 y color2 se crea el color3, donde si los colores a usar son A y G el mensaje debería decir : Con rojo y verde se crea el amarillo. Si alguno de los colores a usar no es primo entonces se deberá mostrar el mensaje No se puede crear un color secundario con la mezcla de un primario y un secundario o de dos secundarios.

Considere que cada color de la paleta usará la primera letra del mismo como identificador exceptuando el caso del Verde que será G y el Azul que será B.

Casos de prueba para verificación de solución

Paleta de colores (posición del color): 1.Rojo, 2.Verde, 3.Azul, 4.Cyan, 5.Amarillo, 6.Violeta

(p.e. el código: 10010 representa G A, verde y amarillo)

Ingrese el código y el número de dígitos: 10010 5

Los colores a usar son: A G

El color A es obtenido por la mezcla de dos colores primarios

El color G es color primario Verde

No se puede crear un color secundario con la mezcla de un primario y un secundario

Casos de prueba para verificación de solución

Paleta de colores (posición del color): 1.Rojo, 2.Verde, 3.Azul, 4.Cyan, 5.Amarillo, 6.Violeta
(p.e. el código: 10010 representa G A, verde y amarillo)
Ingrese el código y el número de dígitos: 110000 6
Los colores a usar son: G R
El color G es color primario Verde
El color R es color primario Rojo
Con rojo y verde se crea el amarillo

2.3.22. Colores en HTML versión 4 (adaptado del laboratorio 5 del ciclo 2020-2)

En HTML un color es construido utilizando una combinación de tres colores primarios, rojo (R), verde(G) y azul(B). Cada uno de estos tres colores puede tomar un valor entre 0 y 255 que representa su participación en la mezcla de colores, siendo 255 la participación máxima y 0 la mínima.

Entonces, un código de color utilizado en HTML se construye con un signo numeral("#") seguido de la intensidad de los colores rojo, verde y azul (en ese orden RGB), escritos en notación hexadecimal. Cuando un color en la escala que va desde 0 a 255 se escribe en notación hexadecimal, se obtiene, siempre, un valor de dos cifras entre 00 y FF. HTML considera mayúsculas y minúsculas, pero para este ejercicio debe considerar solo mayúsculas.

En este esquema, el código de color del rojo más intenso requiere tener el valor "FF" para la parte roja y "00" para las demás, con lo que se obtiene "#FF0000". Asimismo, el código del color verde es "#00FF00" y el del azul "#0000FF". Cuando se mezclan estos tres valores se obtienen resultados equivalentes al de las mezclas en una paleta de pintor, por lo que se puede crear, por ejemplo, amarillo mezclando rojo y verde ("FFFF00"), violeta mezclando rojo y azul ("FF00FF") o cyan mezclando verde y azul ("00FFFF").

Se le pide que utilizando programación modular y en al menos cuatro módulos incluido el principal(main) lea el código de participación de cada color rojo (R), verde(G) y azul(B), a modo de un número de 9 dígitos en base 10, valide que los códigos ingresados sean válidos. De no serlo, deberá mostrar el mensaje según corresponda: si los tres colores son valores inválidos ERROR participación RGB ingresados están equivocados. Si dos de los colores son inválidos ERROR participación RB ingresados inválidos, ERROR participación GB ingresados inválidos, ERROR participación RG ingresados inválidos, según corresponda. Si un color es inválido ERROR participación solo R o G o B es incorrecto. Si los datos son válidos deberá extraer el código de participación de cada color, recuerde que cada código está formado por tres dígitos del código RGB, posteriormente obtener la representación hexadecimal de cada color (el color en base 16) y mostrarlo. Recuerde que debe usar en su solución al menos una estructura iterativa, considere que siempre recibirá un número de 9 dígitos para describir el código RGB. Recuerde que no puede usar estructuras iterativas anidadas.

Casos de prueba para verificación de solución

Ingrese el código RGB:215548983
ERROR GB ingresados inválidos

Casos de prueba para verificación de solución

Ingrese el código RGB:400823215
ERROR RG ingresados inválidos

Casos de prueba para verificación de solución

Ingrese el código RGB:400150889
ERROR RB ingresados inválidos

Casos de prueba para verificación de solución

Ingrese el código RGB:124255180
R=7C
G=FF
B=B4

Casos de prueba para verificación de solución

Ingrese el código RGB:255215000
R=FF
G=D7
B=00

2.3.23. Identificar Cónicas (adaptado del laboratorio 5 del ciclo 2021-1)

Las figuras cónicas son un grupo infinito de formas geométricas clasificadas en cuatro tipos básicos; la circunferencia, la parábola, la elipse y la hipérbola. Ver figura 2.4

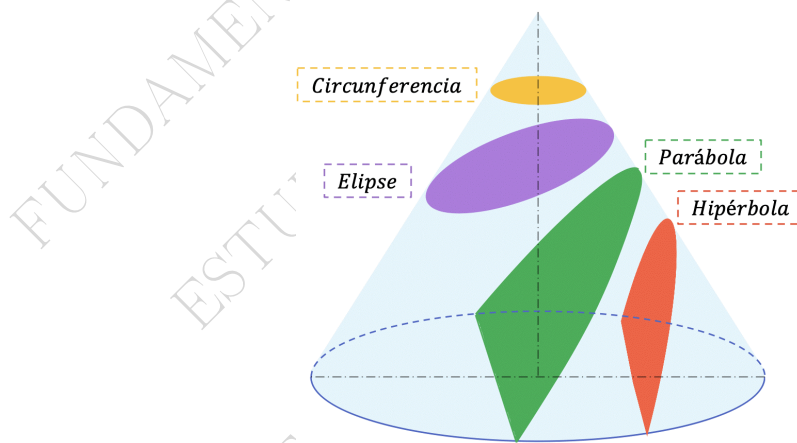


Figura 2.4: Figuras cónicas

Dada la función $f(x, y) = ax^2 + by^2 + cx + dy + e = 0$, donde a , b , c , d y e son números reales.

Se pide que elabore un programa en C que permita al usuario ingresar cierta cantidad de funciones, lea los coeficientes de cada función (valide los coeficientes “a” y “b”, que ambos no sean cero a la vez), en base a la evaluación de los coeficientes “a” y “b” clasifique e imprima el tipo de figura que representa. Finalmente, imprima la cantidad por cada tipo de figuras.

Este programa debe mostrar mensajes específicos ante las siguientes situaciones:

- Al ingresar la cantidad de funciones, debe verificar que esta cantidad tiene que ser mayor que cero. En caso no se cumpla, se deberá emitir el siguiente mensaje “La cantidad de funciones debe ser un número mayor que cero” y el programa debe terminar.
- Al ingresar todos los coeficientes de una función debe verificar que los coeficientes “a” y “b” no pueden ser cero ambos a la vez. En caso no se cumpla, se deberá emitir el siguiente mensaje “Ambos coeficientes a y b no pueden ser igual a cero a la vez” y el programa debe continuar.
- Si el coeficiente “a” es igual cero pero el coeficiente “b” es distinto de cero o el coeficiente “b” es igual cero pero el coeficiente “a” es distinto de cero la figura corresponde a una parábola debiendo imprimir el siguiente mensaje “La figura es una parábola”.
 - Si el coeficiente “a” de la parábola es mayor que cero deberá emitir el siguiente mensaje “La parábola se abre hacia arriba”, en cambio si “a” es menor que cero deberá emitir el siguiente mensaje “La parábola se abre hacia abajo”.
 - Si el coeficiente “b” de la parábola es mayor que cero deberá emitir el siguiente mensaje “La parábola se abre hacia la derecha”, en cambio si “b” es menor que cero deberá emitir el siguiente mensaje “La parábola se abre hacia la izquierda”.
- Si los coeficientes “a” y “b” son iguales la figura corresponde a una circunferencia debiendo imprimir el siguiente mensaje “La figura es una circunferencia”.
- Si los coeficientes “a” y “b” son diferentes y tienen igual signo la figura corresponde a una elipse debiendo imprimir el siguiente mensaje “La figura es una elipse”.
- Si los coeficientes “a” y “b” son diferentes y tienen distinto signo la figura corresponde a una hipérbola debiendo imprimir el siguiente mensaje “La figura es una hipérbola”.

Utilice programación modular (al menos 4 subprogramas incluyendo el main) y como mínimo uno de los módulos debe utilizar parámetros que se modifican luego de la invocación.

Caso de prueba 1

Ingrese la cantidad de funciones: 3

Dada la ecuación cuadrática 1: $ax^2 + by^2 + cx + dy + e = 0$

Ingrese el coeficiente a: 4

Ingrese el coeficiente b: 4

Ingrese el coeficiente c: -12

Ingrese el coeficiente d: 5

Ingrese el coeficiente e: 9

La figura es una circunferencia

Dada la ecuación cuadrática 2: $ax^2 + by^2 + cx + dy + e = 0$

Ingrese el coeficiente a: 0

Ingrese el coeficiente b: 0

Ingrese el coeficiente c: 34

Ingrese el coeficiente d: 12

Ingrese el coeficiente e: 6

Ambos coeficientes a y b no pueden ser igual a cero a la vez

Dada la ecuación cuadrática 3: $ax^2 + by^2 + cx + dy + e = 0$

Ingrese el coeficiente a: 9

Ingrese el coeficiente b: 0

Ingrese el coeficiente c: 12

Ingrese el coeficiente d: -30

Ingrese el coeficiente e: -44

La figura es una parábola

La parábola se abre hacia arriba

Los resultados finales son

Cantidad de parábolas: 1

Cantidad de circunferencias: 1

Cantidad de elipses: 0

Cantidad de hipérbolas: 0

Caso de prueba 2

Ingrese la cantidad de funciones: 5

Dada la ecuación cuadrática 1: $ax^2 + by^2 + cx + dy + e = 0$

Ingrese el coeficiente a: -6

Ingrese el coeficiente b: -6

Ingrese el coeficiente c: 0

Ingrese el coeficiente d: 0

Ingrese el coeficiente e: -18

La figura es una circunferencia

Dada la ecuación cuadrática 2: $ax^2 + by^2 + cx + dy + e = 0$

Ingrese el coeficiente a: -9

Ingrese el coeficiente b: 4

Ingrese el coeficiente c: 12

Ingrese el coeficiente d: -6

Ingrese el coeficiente e: 22

La figura es una hipérbola

Dada la ecuación cuadrática 3: $ax^2 + by^2 + cx + dy + e = 0$

Ingrese el coeficiente a: 0

Ingrese el coeficiente b: 7

Ingrese el coeficiente c: 7

Ingrese el coeficiente d: 0

Ingrese el coeficiente e: 14

La figura es una parábola

La parábola se abre hacia la derecha

Dada la ecuación cuadrática 4: $ax^2 + by^2 + cx + dy + e = 0$

Ingrese el coeficiente a: -4

Ingrese el coeficiente b: -6

Ingrese el coeficiente c: 9

Ingrese el coeficiente d: -7

Ingrese el coeficiente e: 0

La figura es una elipse

Dada la ecuación cuadrática 5: $ax^2 + by^2 + cx + dy + e = 0$

Ingrese el coeficiente a: -11

Ingrese el coeficiente b: 0

Ingrese el coeficiente c: 5

Ingrese el coeficiente d: -21

Ingrese el coeficiente e: 10

La figura es una parábola

La parábola se abre hacia abajo

Los resultados finales son

Cantidad de parábolas: 2

Cantidad de circunferencias: 1

Cantidad de elipses: 1

Cantidad de hipérbolas: 1

Caso de prueba 3

Casos de prueba 3

Ingrese la cantidad de funciones: 0

La cantidad de funciones debe ser un número mayor que cero

2.3.24. Las capas de la esfera (adaptado del laboratorio 5 del ciclo 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 2.5)

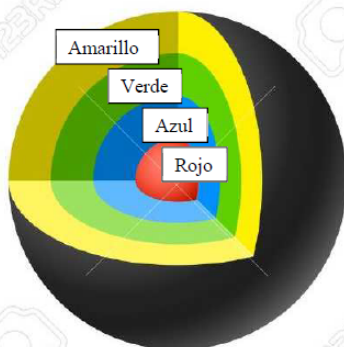


Figura 2.5: Capas

Se pide que elabore un programa en C que permita al usuario ingresar cierta cantidad de puntos en el espacio, para cada uno de ellos se desea calcular la distancia de cada punto al origen de coordenadas e identificar el color de la capa donde está situado dicho punto. Finalmente mostrar la distancia promedio de los puntos, el punto más lejano y el punto más cercano. Además, la cantidad de puntos situados en cada capa.

Cabe señalar que al ingresar la cantidad de puntos, debe verificar que esta cantidad tiene que ser mayor que cero. En caso no se cumpla, se deberá emitir el siguiente mensaje “La cantidad de puntos debe ser un número mayor que cero” y el programa debe terminar.

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 2.6)

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

Figura 2.6: Rango de distancias

Utilice programación modular (al menos 4 subprogramas incluyendo el main) y como mínimo uno de los módulos debe utilizar parámetros que se modifican luego de la invocación.

Caso de prueba 1

Ingrese la cantidad de puntos: -2

La cantidad de puntos debe ser un número mayor que cero

Caso de prueba 2

Ingrese la cantidad de puntos: 3

Ingrese las coordenadas del punto 1

Coordenada x1: 1.00

Coordenada y1: 2.00

Coordenada z1: 3.00

La distancia del punto 1 al origen es: 3.74

El punto 1 cae en el área de color -R

Ingrese las coordenadas del punto 2

Coordenada x2: 5.00

Coordenada y2: 2.00

Coordenada z2: 1.00

La distancia del punto 2 al origen es: 5.48

El punto 2 cae en el área de color -A

Ingrese las coordenadas del punto 3

Coordenada x3: 5.00

Coordenada y3: 5.00

Coordenada z3: 22.00

La distancia del punto 3 al origen es: 23.11

El punto 3 cae en el área de color -V

La distancia promedio es 10.78

El punto 3 (5.00, 5.00, 22.00) es el mas lejano

El punto 1 (1.00, 2.00, 3.00) es el mas cercano

Cantidades de colores

La cantidad de puntos situados en la capa roja es: 1

La cantidad de puntos situados en la capa azul es: 1

La cantidad de puntos situados en la capa verde es: 1

La cantidad de puntos situados en la capa amarilla es: 0

Caso de prueba 3

Ingrese la cantidad de puntos: 5
Ingrese las coordenadas del punto 1
Coordenada x1: 3.00
Coordenada y1: 5.00
Coordenada z1: 28.00
La distancia del punto 1 al origen es: 28.60
El punto 1 cae en el área de color -V

Ingrese las coordenadas del punto 2
Coordenada x2: 25.00
Coordenada y2: 39.00
Coordenada z2: 9.00
La distancia del punto 2 al origen es: 47.19
El punto 2 cae en el área de color -M

Ingrese las coordenadas del punto 3
Coordenada x3: 38.00
Coordenada y3: 13.00
Coordenada z3: 5.00
La distancia del punto 3 al origen es: 40.47
El punto 3 cae en el área de color -M

Ingrese las coordenadas del punto 4
Coordenada x4: 5.00
Coordenada y4: 5.00
Coordenada z4: 22.00
La distancia del punto 4 al origen es: 23.11
El punto 4 cae en el área de color -V

Ingrese las coordenadas del punto 5
Coordenada x5: 9.00
Coordenada y5: 10.00
Coordenada z5: 6.00
La distancia del punto 5 al origen es: 14.73
El punto 5 cae en el área de color -A

La distancia promedio es 30.82
El punto 2 (25.00, 39.00, 9.00) es el mas lejano
El punto 5 (9.00, 10.00, 6.00) es el mas cercano

Cantidades de colores
La cantidad de puntos situados en la capa roja es: 0
La cantidad de puntos situados en la capa azul es: 1
La cantidad de puntos situados en la capa verde es: 2
La cantidad de puntos situados en la capa amarilla es: 2

2.3.25. Las sumatorias (adaptado del laboratorio 5 del ciclo 2021-1)

Se pide que elabore un programa en C que permita al usuario ingresar una letra “A” o “B” (también podría ingresar la letra en minúscula) para elegir una opción de cálculo de una determinada sumatoria.

Si ingresa la opción “A” o “a” debe realizar la sumatoria de la figura 2.7

$$= \frac{a}{2} - \frac{3a^2}{4} + \frac{5a^4}{6} - \frac{7a^8}{8} + \frac{9a^{16}}{10} - \dots (-1)^{n+1} \frac{(2n-1)a^{2^{n-1}}}{2n}$$

Figura 2.7: Sumatoria 1

Donde:

- La cantidad de términos está representada por la n que pertenece al siguiente rango $0 < n < 21$.
- a es un número real entre 0.5 y 2.
- Sumatoria 1 es el resultado de sumar los n términos.

Si ingresa la “B” o “b” debe realizar la sumatoria de la figura 2.8

$$= \frac{1}{X^n} - \frac{2}{X^{n-1}} + \frac{3}{X^{n-2}} - \frac{4}{X^{n-3}} + \dots + \frac{n}{X}$$

Figura 2.8: Sumatoria 2

Donde:

- La cantidad de términos está representada por la n
- n y x son números reales positivos.
- Sumatoria 2 es el resultado de sumar los n términos.

Este 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 la cantidad de términos (n), a y x , según corresponda y mostrar el mensaje de error respectivo (ver casos de prueba).

Utilice programación modular (al menos 4 subprogramas incluyendo el main) y como mínimo uno de los módulos debe utilizar parámetros que se modifican luego de la invocación.

Caso de prueba 1

Elija una opción A o B: a
 Ingrese la cantidad de términos: 30
 Ingrese el valor de a: 1
 Error en la cantidad de términos

Caso de prueba 2

Elija una opción A o B: B
 Ingrese la cantidad de términos: 10
 Ingrese el valor de x: -2
 Error en el valor de x

Caso de prueba 3

Elija una opción A o B: A
 Ingrese la cantidad de términos: -5
 Ingrese el valor de a: 3
 Error en la cantidad de términos y el valor de a

Caso de prueba 4

Elija una opción A o B: A
 Ingrese la cantidad de términos: 5
 Ingrese el valor de a: 1
 El resultado de la sumatoria 1 es: 0.6083

Caso de prueba 5

Elija una opción A o B: b
 Ingrese la cantidad de términos: 5
 Ingrese el valor de x: 5
 El resultado de la sumatoria 2 es: 0.8611

Caso de prueba 6

Elija una opción A o B: D
 Error en la opción ingresada

2.3.26. Más sumatorias (adaptado del laboratorio 5 del ciclo 2021-1)

Se pide que elabore un programa en C que permita al usuario ingresar una letra “S” o “T” (también podría ingresar la letra en minúscula) para elegir una opción de cálculo de una determinada sumatoria.

Si ingresa la opción “S” o “s” debe realizar el cálculo que se indica en la figura 2.9

$$S = \sum_{k=1}^n (-1)^{k+1} \frac{a^k b^{k+1}}{(a-b)(k)}$$

Figura 2.9: Sumatoria 1

Si $n = 10$ entonces S se calcularía como la figura 2.10

$$S = (-1)^{1+1} \frac{a^1 b^{1+1}}{(a-b)(1)} + (-1)^{2+1} \frac{a^2 b^{2+1}}{(a-b)(2)} + (-1)^{3+1} \frac{a^3 b^{3+1}}{(a-b)(3)} + \dots + (-1)^{10+1} \frac{a^{10} b^{10+1}}{(a-b)(10)}$$

Figura 2.10: Sumatoria 1 para $n=10$

Donde:

- La cantidad de términos está representada por la n que pertenece al siguiente rango $1 \leq n \leq 10$
- a es un número entero entre 4 y 6.
- b es un número entero entre 1 y 3.

- S es el resultado de sumar los n términos.

Si ingresa la opción “T” o “t” debe realizar el cálculo que se indica en la figura 2.11

$$= -3x + \frac{6y}{2a} - \frac{9x}{4a^2} + \frac{12y}{8a^3} - \frac{15x}{16a^4} + \dots$$

Figura 2.11: Sumatoria 2

Donde:

- La cantidad de términos está representada por la n que pertenece al siguiente rango $1 \leq n \leq 30$
- a, x e y son números reales positivos
- T es el resultado de sumar los n términos.

Este programa debe mostrar mensajes específicos ante las siguientes situaciones:

- Al ingresar la opción debe verificar que sea “S” o “T” (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 la cantidad de términos (n), a, b, x e y, según corresponda y mostrar el mensaje de error respectivo (ver casos de prueba)

Utilice programación modular (al menos 4 subprogramas incluyendo el main) y como mínimo uno de los módulos debe utilizar parámetros que se modifican luego de la invocación.

Caso de prueba 1

Elija una opción S o T: s
 Ingrese la cantidad de términos: 10
 Ingrese el valor de a: 5
 Ingrese el valor de b: 2
 El resultado de la sumatoria S es: -600072793.8624

Caso de prueba 2

Elija una opción S o T: T
 Ingrese la cantidad de términos: 5
 Ingrese el valor de a: 2
 Ingrese el valor de x: 1
 Ingrese el valor de y: 1
 El resultado de la sumatoria T es: -1.9336

Caso de prueba 3

Elija una opción S o T: E
 Error en la opción ingresada

Caso de prueba 4

Elija una opción S o T: S
 Ingrese la cantidad de términos: 5
 Ingrese el valor de a: 10
 Ingrese el valor de b: 5
 Error en los coeficientes a y/o b

Caso de prueba 5

Elija una opción S o T: s
 Ingrese la cantidad de términos: -10
 Ingrese el valor de a: 0
 Ingrese el valor de b: 5
 Error en la cantidad de términos y en los coeficientes a y/o b

Caso de prueba 6

Elija una opción S o T: t
 Ingrese la cantidad de términos: 12
 Ingrese el valor de a: -2
 Ingrese el valor de x: 0
 Ingrese el valor de y: -3
 Error en los coeficientes y, x y/o a

2.3.27. La parábola (adaptado del laboratorio 5 del ciclo 2021-1)

La parábola es el lugar geométrico de los puntos del plano que equidistan de un punto fijo llamado foco y de una recta fija llamada directriz. La fórmula es la siguiente: $f(x) = ax^2 + bx + c$, donde a, b, y c son números reales y a es diferente de cero.

La gráfica de f es una parábola con vértice (h,k). Se conoce que la parábola se abre hacia arriba si $a > 0$ o hacia abajo si $a < 0$ tal como se muestra en la figura 2.12

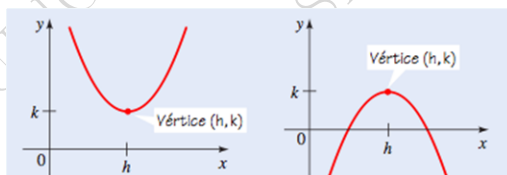


Figura 2.12: Parábola

Para calcular el vértice de una parábola se utiliza la siguiente fórmula:

$$V\left(\frac{-b}{2a}, \frac{4ac-b^2}{4a}\right)$$

Se pide que elabore un programa en C que permita al usuario ingresar cierta cantidad de parábolas, lea los coeficientes de la ecuación cada parábola (valide que el coeficiente “a” sea diferente de cero), calcule el vértice de cada una, indique si se abre hacia arriba o hacia abajo e imprima en que cuadrante está cada vértice. Finalmente, encuentre el vértice que se encuentra más a la derecha y el vértice que se encuentra más abajo. Este programa debe mostrar mensajes específicos ante las siguientes situaciones:

- Al ingresar la cantidad de parábolas, debe verificar que esta cantidad tiene que ser mayor que cero. En

caso no se cumpla, se deberá emitir el siguiente mensaje “La cantidad de parábolas debe ser un número mayor que cero” y el programa debe terminar.

- Al ingresar todos los coeficientes de una parábola debe verificar que el coeficiente “a” tiene que ser diferente de cero. En caso no se cumpla, se deberá emitir el siguiente mensaje “El coeficiente a no puede ser igual a cero” y el programa debe continuar.
- Si el coeficiente “a” de la parábola es mayor que cero deberá emitir el siguiente mensaje “La parábola se abre hacia arriba”, en cambio si “a” es menor que cero deberá emitir el siguiente mensaje “La parábola se abre hacia abajo”.
- Si las coordenadas del vértice (h,k) están dentro del primer cuadrante del eje de coordenadas deberá emitir el siguiente mensaje “El vértice está en el primer cuadrante”, si está dentro del segundo cuadrante deberá emitir el mensaje “El vértice está en el segundo cuadrante” y así sucesivamente con el resto de cuadrantes. También podría ser que el caso que el vértice coincida con el origen de coordenadas para ello deberá emitir el siguiente mensaje “El vértice está en el origen de coordenadas”. Si el vértice está sobre el eje y debe emitir el siguiente mensaje “El vértice está en el eje y” y si el vértice está sobre el eje x debe emitir el siguiente mensaje “El vértice está en el eje x”.

Utilice programación modular (al menos 4 subprogramas incluyendo el main) y como mínimo uno de los módulos debe utilizar parámetros que se modifican luego de la invocación.

Caso de prueba 1

Ingrese la cantidad de parábolas: 3

Dada la ecuación cuadrática 1 de una parábola $ax^2 + bx + c = 0$

Ingrese el coeficiente a: -1

Ingrese el coeficiente b: 0

Ingrese el coeficiente c: 0

El vértice de la parábola es: (0.00,0.00)

La parábola se abre hacia abajo

El vértice está en el origen de coordenadas

Dada la ecuación cuadrática 2 de una parábola $ax^2 + bx + c = 0$

Ingrese el coeficiente a: 1

Ingrese el coeficiente b: 3

Ingrese el coeficiente c: -2

El vértice de la parábola es: (-1.50,-4.25)

La parábola se abre hacia arriba

El vértice está en el tercer cuadrante

Dada la ecuación cuadrática 3 de una parábola $ax^2 + bx + c = 0$

Ingrese el coeficiente a: 1

Ingrese el coeficiente b: -2

Ingrese el coeficiente c: 1

El vértice de la parábola es: (1.00,0.00)

La parábola se abre hacia arriba

El vértice está en el eje x

El vértice que se encuentra más a la derecha del origen es (1.00,0.00)

El vértice que se encuentra más a abajo del origen es (-1.50,-4.25)

Caso de prueba 2

Ingrese la cantidad de parábolas: 3

Dada la ecuación cuadrática 1 de una parábola $ax^2 + bx + c = 0$

Ingrese el coeficiente a: 1

Ingrese el coeficiente b: 4

Ingrese el coeficiente c: 8

El vértice de la parábola es: (-2.00,4.00)

La parábola se abre hacia arriba

El vértice está en el segundo cuadrante

Dada la ecuación cuadrática 2 de una parábola $ax^2 + bx + c = 0$

Ingrese el coeficiente a: 0

Ingrese el coeficiente b: -2

Ingrese el coeficiente c: 8

El coeficiente a no puede ser igual a cero

Dada la ecuación cuadrática 3 de una parábola $ax^2 + bx + c = 0$

Ingrese el coeficiente a: -4

Ingrese el coeficiente b: -12

Ingrese el coeficiente c: 20

El vértice de la parábola es: (-1.50,29.00)

La parábola se abre hacia abajo

El vértice está en el segundo cuadrante

El vértice que se encuentra más a la derecha del origen es (-1.50, 29.00)

El vértice que se encuentra más a abajo del origen es (-2.00,4.00)

Caso de prueba 3

Ingrese la cantidad de parábolas: 0

La cantidad de parábolas debe ser un número mayor que cero