

FUNDAMENTOS DE PROGRAMACIÓN
LABORATORIO 6
PROPUESTAS DE SOLUCIÓN
SEMESTRE ACADÉMICO 2021-2

Horarios: Todos los horarios

Elaborado por Jorge Berrocal

INDICACIONES:

- Debe utilizar variables descriptivas, comentarios y mensajes descriptivos.
- El orden y la eficiencia de su implementación serán considerados en la calificación.

RESULTADOS ESPERADOS:

- Al finalizar la sesión, el alumno comprenderá el funcionamiento de la estructura algorítmica selectiva múltiple.
- Al finalizar la sesión, el alumno construirá programas usando estructuras algorítmicas selectivas múltiple.
- Al finalizar la sesión, el alumno comprenderá el funcionamiento de la estructura algorítmica iterativa con salida controlada.
- Al finalizar la sesión, el alumno construirá programas usando la estructura algorítmica iterativa con salida controlada.

CONSIDERACIONES:

- La solución presentada para cada problema corresponde a una propuesta de solución por parte del autor.
- En programación pueden existir muchas soluciones para un mismo problema pero debe cumplir con todo lo solicitado, incluyendo las restricciones brindadas.

Desarrolle los siguientes problemas en lenguaje C:

1. Aplicaciones de la Razón Aurea

El Número Áureo, también conocido como la Razón Aurea, se trata de un número irracional que posee muchas propiedades interesantes vinculadas con la belleza de las cosas. La Razón Aurea fue descubierta en la Antigüedad, como una relación o proporción entre dos segmentos de una recta, es decir, una construcción geométrica.

Existen diferentes figuras geométricas que pueden ser construidas con la razón aurea, estas figuras tienen la característica que, por su proporcionalidad ante el espectador los objetos construidos con sus dimensiones, realzan la belleza de las cosas.

Tres ejemplos de figuras áureas son:

- **Rectángulo Áureo.-** Es el rectángulo que cumple con que su altura (lado mayor) es igual a la base (lado menor) por la razón aurea.
- **Triángulo Áureo.-** Es un triángulo isósceles que cumple con que cada uno de sus lados iguales son igual a la base (lado diferente) por la razón aurea.
- **Trapezio Áureo.-** Es un trapezio isósceles que cumple con que su base mayor es igual a uno de los lados iguales por la razón aurea mientras que la base menor es igual a uno de los lados iguales entre la razón aurea.

$$Lado Mayor = Razón Aurea * Lado Menor$$

Figura 1: Fórmula del Rectángulo Aureo

$$Lado Igual = Razón Aurea * Base$$

Figura 2: Fórmula del Triángulo Áureo

$$Base Mayor = Razón Aurea * Lado Igual \quad y \quad Base Menor = \frac{Lado Igual}{Razón Aurea}$$

Figura 3: Fórmulas del Trapecio Aureo

La Razón Áurea se puede aproximar a través de la siguiente expresión matemática:

$$Razón Aurea (\varphi) = \sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + \dots}}}}}}$$

Figura 4: Expresión de la Razón Aurea

Se pide desarrollar un programa en Lenguaje C que permita leer el número de iteraciones con las que se calculará la aproximación de la Razón Aurea (deberá validar que dicho número sea mayor o igual a 10), y después solicite el ingreso de un carácter que indique el tipo de figura aurea con la que se trabajará (R para un rectángulo áureo, T para triángulo áureo o P para un trapecio áureo). Considerar que en caso se ingrese un caracter diferente deberá indicar un mensaje de error.

En caso haber seleccionado una letra R, deberá solicitar la base del rectángulo, debiéndose calcular su altura. En caso haber seleccionado una letra T, deberá solicitar el ingreso de la base del triángulo áureo (lado diferente), debiéndose calcular su altura, así como los lados iguales. Por último, si seleccionó una letra P, deberá solicitar el ingreso del uno de los lados laterales (lados iguales), para calcular las dos bases de dicho trapecio.

La solución deberá contar, además del módulo principal, con un módulo para calcular la razón aurea. Este módulo deberá recibir como parámetro un número con la cantidad de iteraciones a utilizar para la aproximación.

Caso de prueba 1

Ingrese el número de iteraciones para calcular la razón aurea: 5
El número de iteraciones debe ser mayor o igual a 10.

Caso de prueba 2

Ingrese el número de iteraciones para calcular la razón aurea: 150
Ingrese el tipo de objeto (Rectángulo [R], Triángulo [T] o Trapecio [P]): C
El tipo de objeto ingresado no es válido.

Caso de prueba 3

Ingrese el número de iteraciones para calcular la razón aurea: 100
Ingrese el tipo de objeto (Rectángulo [R], Triángulo [T] o Trapecio [P]): R
Ingrese la base del rectángulo áureo: 25
La altura de un rectángulo áureo de base 25.00 es 40.45

Caso de prueba 4

Ingrese el número de iteraciones para calcular la razón aurea: 150
Ingrese el tipo de objeto (Rectángulo [R], Triángulo [T] o Trapecio [P]): T
Ingrese la base del triángulo áureo: 30
La altura de un triángulo áureo de base 30.00 es 46.17 y sus lados iguales son de 48.54

Caso de prueba 5

Ingrese el número de iteraciones para calcular la razón aurea: 200
Ingrese el tipo de objeto (Rectángulo [R], Triángulo [T] o Trapecio [P]): P
Ingrese el lado lateral del trapecio áureo: 25
Las bases de un trapecio áureo de lado lateral 25.00 son 40.45 y 15.45

Programa 1: Propuesta de solución - Aplicaciones de la razón aurea

```
1  #include <stdio.h>
2  #include <math.h>
3
4  double CalcularRazon_Aurea(int);
5
6  int main ()
7  {
8      char Tipo_Objeto;
9      int Cant_Iteraciones;
10     double Razon_Aurea, Base, Altura, LadoMayor, LadoLateral, BaseMayor, BaseMenor;
11
12     //Consulta el número de iteraciones de la aproximación.
13     printf("Ingrese el numero de iteraciones para calcular la razon aurea: ");
14     scanf("\n%d", &Cant_Iteraciones);
15
16     if ( Cant_Iteraciones>=10 )
17     {
18         //Consulta el tipo de objeto a evaluar.
19         printf("Ingrese el tipo de objeto (Rectangulo [R], Triangulo [T] o Trapecio [P]): ");
20         scanf("\n%c", &Tipo_Objeto);
21
22         //Calcula la Razón Aurea con un número dado de iteraciones.
23         Razon_Aurea = CalcularRazon_Aurea (Cant_Iteraciones);
24
25         switch ( Tipo_Objeto )
26         {
27             case 'R' :
28                 printf("\nIngrese la base del rectangulo aureo: ");
29                 scanf("%lf", &Base);
30
31                 //Obtiene la altura de un rectángulo aureo e imprime el resultado.
32                 Altura = Razon_Aurea * Base;
33                 printf("La altura de un rectangulo aureo de base %.2lf es %.2lf", Base, Altura);
34                 break;
35             case 'T' :
36                 printf("\nIngrese la base del triangulo aureo: ");
37                 scanf("%lf", &Base);
```

```

38                                     //Obtiene el lado mayor del triangulo aureo e imprime el resultado.
39                                     LadoMayor = Razon_Aurea * Base;
40
41                                     //Obtiene la Altura del triángulo aureo.
42                                     Altura = sqrt( pow(LadoMayor, 2) - pow(Base/2, 2) );
43                                     printf("La altura de un triangulo aureo de base %.2lf es %.2lf y sus lados iguales son de %.2lf",
44                                             Base, Altura, LadoMayor);
45                                     break;
46                             case 'P' :
47                                 printf("\nIngrese el lado lateral del trapecio aureo: ");
48                                 scanf("%lf", &LadoLateral);
49
50                                 //Obtiene las bases del trapecio aureo e imprime los resultados;
51                                 BaseMayor = LadoLateral * Razon_Aurea;
52                                 BaseMenor = LadoLateral / Razon_Aurea;
53                                 printf("Las bases de un trapecio aureo de lado lateral %.2lf son %.2lf y %.2lf", LadoLateral,
54                                         BaseMayor, BaseMenor);
55                                 break;
56                             default :
57                                 printf("El tipo de objeto ingresado no es valido.");
58                         }
59                     else
60                         printf("El numero de iteraciones debe ser mayor o igual a 10.");
61             }
62
63             double CalcularRazon_Aurea (int Cant_Iteraciones)
64             {
65                 int Cont = 1;
66                 double Razon_Aurea = 1; //Puede inicializarse con 0 o 1.
67
68                 //Ejecuta cada iteración que calcula la Razón Aurea.
69                 do
70                 {
71                     Razon_Aurea = sqrt(1 + Razon_Aurea);
72
73                     Cont++;
74
75                 } while (Cont <= Cant_Iteraciones);
76
77                 //Devuelve la Razón Aurea calculada.
78                 return Razon_Aurea;
79             }

```

2. Aplicaciones del Número e

El Número e es uno de los números irracionales más importantes por su aplicación en diversas ramas de las Matemáticas, al ser la base de los logaritmos naturales y formar parte de las ecuaciones del interés compuesto y muchos otros problemas.

Un par de ejemplos de la aplicación del Número e son:

- **La estimación del tamaño de una población** después de transcurrido un número dado de años, para lo cual la Población Final cumplirá la siguiente relación:
- **El cálculo del interés compuesto continuo**, con el cual es posible calcular cuánto será el total ahorrado en un depósito, dada una Tasa de Interés después de un número de años dado, para lo cual el ahorro final cumplirá la siguiente relación:
- Para trabajar con ecuaciones donde la variable (x) a despejar se encuentre dentro de un **logaritmo natural** con la siguiente estructura:

$$Población\ Final = Población\ Actual * e^{\left(\frac{Tasa\ Crecimiento}{100} * Número\ Años\right)}$$

Figura 5: Estimación de la Población

$$Ahorro\ Final = Ahorro\ Inicial * e^{\left(\frac{Tasa\ Interés}{100} * Número\ Años\right)}$$

Figura 6: Cálculo del Ahorro Final

El Número e se puede aproximar a través de la siguiente expresión matemática:

Se pide desarrollar un programa en Lenguaje C que permita leer el número de iteraciones con las que se calculará la aproximación del Número e (deberá validar que dicho número sea mayor o igual a 10) y después solicite el ingreso de un caracter que indique el tipo de aplicación con la que se trabajará (P para crecimiento poblacional, I para el cálculo del interés compuesto continuo o L para el despeje del Logaritmo Natural). Considerar que en caso se ingrese un caracter diferente deberá indicar un mensaje de error.

En caso haber seleccionado una letra P, deberá solicitar la población actual, la tasa de crecimiento poblacional y el tiempo en años de la estimación, debiéndose calcular la población final estimada. En caso haber seleccionado una letra I, deberá solicitar el ahorro inicial, la tasa de interés y el número de años a ahorrar, debiéndose calcular el ahorro final al término de los años indicados. Por último en caso seleccionar una letra L deberá solicitar el ingreso de una constante con la que despejar la ecuación $\ln(x) = \text{Constante}$.

La solución deberá contar, además del módulo principal, con un módulo para calcular el Número e. Este módulo deberá recibir como parámetro un número con la cantidad de iteraciones a utilizar para la aproximación.

Caso de prueba 1

Ingrese el número de iteraciones para calcular el Numero e: 9
El número de iteraciones debe ser mayor o igual a 10.

Caso de prueba 2

Ingrese el número de iteraciones para calcular el Numero e: 50
Ingrese el tipo de aplicación (Crecimiento Poblacional [P], Interés Compuesto [I], Logaritmo Natural [L]):
X
El tipo de aplicación no es válido.

$$\ln(x) = \text{Constante} \implies x = e^{\text{Constante}}$$

Figura 7: Despeje de una variable con Logaritmo Natural

$$e = 1 + \frac{1}{1} + \frac{1}{1 * 2} + \frac{1}{1 * 2 * 3} + \frac{1}{1 * 2 * 3 * 4} + \dots$$

Figura 8: Expresión del Número e

Caso de prueba 3

Ingrese el número de iteraciones para calcular el Numero e: 100
 Ingrese el tipo de aplicación (Crecimiento Poblacional [P], Interés Compuesto [I], Logaritmo Natural [L]):
 P
 Ingrese la población actual: 33000000
 Ingrese la tasa de crecimiento poblacional anual (0 a 100): 3
 Ingrese el tiempo a transcurrir en años: 10 La población final estimada a los 10.00 será de 44545341 personas

Caso de prueba 4

Ingrese el número de iteraciones para calcular el Numero e: 100
 Ingrese el tipo de aplicación (Crecimiento Poblacional [P], Interés Compuesto [I], Logaritmo Natural [L]):
 I
 Ingrese el dinero depositado en soles: 500000
 Ingrese la tasa de interés anual (0 a 100): 4
 Ingrese el tiempo del depósito en años: 10
 El ahorro al finalizar los 10.00 años será de 745912.35 soles

Caso de prueba 5

Ingrese el número de iteraciones para calcular el Numero e: 50
 Ingrese el tipo de aplicación (Crecimiento Poblacional [P], Interés Compuesto [I], Logaritmo Natural [L]):
 L
 Ingrese la constante a evaluar para despejar la variable X: 2
 Para una ecuación de la forma $\ln(X) = 2.00$ el valor de X que la resuelve es 7.39

Programa 2: Propuesta de solución - Aplicaciones del número e

```

1  #include <stdio.h>
2  #include <math.h>
3
4  double Calcular_Numero_e(int);
5
6  int main ()
7  {
8      char Tipo_Aplicacion;
9      int Cant_Iteraciones;
10     double Numero_e, PoblacionActual, PoblacionFinal, TasaCrecimientoPoblacional, Tiempo;
11     double AhorroInicial, AhorroFinal, TasaInteres, Constante, Variable;
12
13     //Consulta el número de iteraciones para el cálculo de la aproximación del Número e.
14     printf("Ingrese el numero de iteraciones para calcular el Numero e: ");
  
```

```

15 scanf("\n%d", &Cant_Iteraciones);
16
17 if ( Cant_Iteraciones>10 )
18 {
19     //Calcula el Número e con un número dado de iteraciones.
20     Numero_e = Calcular_Numero_e(Cant_Iteraciones);
21
22     printf("Ingrese el tipo de aplicacion\n");
23     printf("Crecimiento Poblacional [P], Interes Compuesto [I], Logaritmo Natural [L]: ");
24     scanf("\n%c", &Tipo_Aplicacion);
25
26     switch ( Tipo_Aplicacion )
27     {
28         case 'P' :
29             //Solicita los datos para el cálculo de la población final.
30             printf("\nIngrese la poblacion actual: ");
31             scanf("%lf", &PoblacionActual);
32             printf("Ingrese la tasa de crecimiento poblacional anual (0 a 100): ");
33             scanf("%lf", &TasaCrecimientoPoblacional);
34             printf("Ingrese el tiempo a transcurrir en anos: ");
35             scanf("%lf", &Tiempo);
36
37             //Estima la población final al término de un tiempo dado.
38             PoblacionFinal = PoblacionActual * pow(Numero_e, (TasaCrecimientoPoblacional/100)*Tiempo);
39             printf("La poblacion final estimada a los %.2lf sera de %.0lf personas", Tiempo, PoblacionFinal);
40             break;
41         case 'I' :
42             //Solicita los datos para el cálculo del ahorro al final del periodo.
43             printf("\nIngrese el dinero depositado en soles: ");
44             scanf("%lf", &AhorroInicial);
45             printf("Ingrese la tasa de interes anual (0 a 100): ");
46             scanf("%lf", &TasaInteres);
47             printf("Ingrese el tiempo del deposito en anos: ");
48             scanf("%lf", &Tiempo);
49
50             //Estima la población final al término de un tiempo dado.
51             AhorroFinal = AhorroInicial * pow(Numero_e, (TasaInteres/100)*Tiempo);
52             printf("El ahorro al finalizar los %.2lf anos sera de %.2lf soles", Tiempo, AhorroFinal);
53             break;
54         case 'L' :
55             //Solicita la constante para despejar el logaritmo natural.
56             printf("\nIngrese la constante a evaluar para despejar la variable X: ");
57             scanf("%lf", &Constante);
58
59             //Calcula el valor de la Variable e imprime el resultado.
60             Variable = pow(Numero_e, Constante);
61             printf("Para una ecuacion de la forma Ln(X) = %.2lf el valor de X que la resuelve es %.2lf",
62                 Constante, Variable);
63             break;
64         default :
65             printf("El tipo de aplicacion no es valido.");
66     }
67     else
68         printf("El numero de iteraciones debe ser mayor o igual a 10.");
69 }
70
71 double Calcular_Numero_e(int Cant_Iteraciones)
72 {
73     int Cont=1; //Se inicializa el contador en 1.
74     double Producto, Numero_e;
75
76     //Se inicializan las variables para el cálculo del exponencial.
77     Producto = 1;
78     Numero_e = 1;
79
80     //Ejecuta cada iteración que calcula la Razón Aurea.

```

```

81     do
82     {
83         //Acumula el contador en Producto
84         Producto = Producto * Cont;
85
86         //Obtiene el valor del Número e.
87         Numero_e = Numero_e + pow(Producto, -1);
88
89         Cont++;
90
91     } while (Cont <= Cant.Iteraciones);
92
93     //Devuelve el exponencial calculado.
94     return Numero_e;
95 }

```

3. Aplicaciones del Número Pi

El Número Pi es uno de los números irracionales más importantes por su aplicación en diversas ramas de las Matemáticas. Algunos ejemplos de la aplicación del Número Pi son:

- El cálculo del **periodo de oscilación** de un péndulo (ver figura 9).

$$Periodo\ de\ Oscilación\ (seg) = 2\pi \sqrt{\frac{Longitud\ del\ Péndulo\ (m)}{Aceleración\ del\ Péndulo\ (m/Seg^2)}}$$

Figura 9: Fórmula del periodo de oscilación de un Péndulo

- El cálculo del **área lateral de la superficie** de un cono (ver figura 10).

$$Área\ Lateral = \pi * Radio\ de\ la\ base\ (m) * Longitud\ del\ lado\ (m)$$

Figura 10: Fórmula del área lateral de un Cono

- El cálculo del **área de una elipse** (ver figura 11).

$$Área\ Elipse = \pi * Semi\ Eje\ Mayor\ (m) * Semi\ Eje\ Menor\ (m)$$

Figura 11: Fórmula del área de una Elipse

El Número Pi se puede aproximar a través de la siguiente expresión matemática:

Se pide desarrollar un programa en Lenguaje C que permita leer el número de iteraciones con las que se calculará la aproximación del Número Pi (deberá validar que dicho número sea mayor o igual a 5 y menor igual que 20) y después solicite el ingreso de un caracter que indique el tipo de aplicación con la que se trabajará el valor de Pi (P para el cálculo del periodo de oscilación de un péndulo, C para el cálculo del área lateral de superficie de un cono o E para el área de una elipse). Considerar que en caso se ingrese un caracter diferente deberá indicar un mensaje

$$\pi = 2^n * X_n$$

Sabiendo que X_n es el enésimo término de la siguiente serie:

$$X_n = \sqrt{2 - 2\sqrt{1 - \frac{X_{n-1}^2}{4}}} \quad \text{y} \quad x_1 = \sqrt{2}$$

Figura 12: Expresión de la aproximación de Pi

de error.

En caso haber seleccionado una letra P, deberá solicitar la longitud del péndulo y la aceleración del péndulo para calcular la longitud del péndulo. En caso haber seleccionado una letra C, deberá solicitar el radio de la base del cono y la longitud del lado del cono, para calcular el área lateral de superficie del cono. Por último, en caso haya seleccionado una letra E deberá pedir las longitudes de los semiejes mayor y menor para calcular el área de la elipse.

La solución deberá contar, además del módulo principal, con un módulo para calcular el Número Pi. Este módulo deberá recibir como parámetro un número con la cantidad de iteraciones a utilizar para la aproximación.

Caso de prueba 1

Ingrese el número de iteraciones para calcular Pi: 40
El número de iteraciones debe ser mayor a 5 y menor o igual a 20.

Caso de prueba 2

Ingrese el número de iteraciones para calcular Pi: 15
Ingrese el tipo de aplicación del Número Pi
(Periodo del péndulo [P], Área lateral de superficie de un cono [C] o Área de una Elipse [E]): D
La aplicación ingresada no es válida.

Caso de prueba 3

Ingrese el número de iteraciones para calcular Pi: 20
Ingrese el tipo de aplicación del Número Pi
(Periodo del péndulo [P], Área lateral de superficie de un cono [C] o Área de una Elipse [E]): P
Ingrese la longitud del péndulo (metros): 0.30
Ingrese la aceleración del péndulo (metros/segundo²): 9.8
El periodo de un péndulo de longitud 0.30 y aceleración 9.80 es 1.10 segundos

Caso de prueba 4

Ingrese el número de iteraciones para calcular Pi: 15

Ingrese el tipo de aplicación del Número Pi

(Periodo del péndulo [P], Área lateral de superficie de un cono [C] o Área de una Elipse [E]): C

Ingrese el radio de la base del cono (metros): 12

Ingrese el lado lateral del cono (metros): 15

El área lateral de un cono de radio 12.00 en la base y lado lateral 15.00 es 565.49 metros²

Caso de prueba 5

Ingrese el número de iteraciones para calcular Pi: 18

Ingrese el tipo de aplicación del Numero Pi

(Periodo del péndulo [P], Área lateral de superficie de un cono [C] o Área de una Elipse [E]): E

Ingrese la longitud del semieje mayor de la elipse (metros): 5

Ingrese la longitud del semieje menor de la elipse (metros): 4

El área de una elipse de semiejes 5.00 y 4.00 es 62.83 metros²

Programa 3: Propuesta de solución - Aplicaciones del número PI

```
1  #include <stdio.h>
2  #include <math.h>
3
4  double CalcularNumeroPi(int);
5
6  int main ()
7  {
8      char Tipo_Aplicacion;
9      int Cant_Iteraciones;
10     double NumeroPi, Longitud, Aceleracion, PeriodoPendulo, Radio, LadoLateral, AreaLateral;
11     double SemiEjeMayor, SemiEjeMenor, AreaElipse;
12
13     //Consulta el número de iteraciones de la aproximación.
14     printf("Ingrese el numero de iteraciones para calcular Pi: ");
15     scanf("%d", &Cant_Iteraciones);
16
17     if ( Cant_Iteraciones >= 5 && Cant_Iteraciones <= 20 )
18     {
19         //Consulta el tipo de objeto a evaluar.
20         printf("Ingrese el tipo de aplicacion del Numero Pi, \n");
21         printf("(Periodo del pendulo [P], Area lateral de superficie de un cono [C] o Area de una Elipse [E]): ");
22         scanf("%c", &Tipo_Aplicacion);
23
24         //Calcula la Razón Aurea con un número dado de iteraciones.
25         NumeroPi = CalcularNumeroPi (Cant_Iteraciones);
26
27         switch ( Tipo_Aplicacion )
28         {
29             case 'P' :
30                 //Solicita los datos de la aplicación
31                 printf("\nIngrese la longitud del pendulo (metros): ");
32                 scanf("%lf", &Longitud);
33                 printf("Ingrese la aceleracion del pendulo (metros/segundo^2): ");
34                 scanf("%lf", &Aceleracion);
35
36                 //Obtiene el periodo del péndulo e imprime el resultado.
37                 PeriodoPendulo = 2 * NumeroPi * sqrt(Longitud/Aceleracion);
38                 printf("El periodo de un pendulo de longitud %.2lf y aceleracion %.2lf es %.2lf segundos",
39                     Longitud, Aceleracion, PeriodoPendulo);
40                 break;
41             case 'C' :
```

```

42 //Solicita los datos de la aplicación
43 printf("\nIngrese el radio de la base del cono (metros): ");
44 scanf("%lf", &Radio);
45 printf("Ingrese el lado lateral del cono (metros): ");
46 scanf("%lf", &LadoLateral);
47
48 //Obtiene el área lateral del cono e imprime el resultado.
49 AreaLateral = NumeroPi * Radio * LadoLateral;
50 printf("El area lateral de un cono de radio %.2lf en la base y lado lateral %.2lf es %.2lf metros2",
51       Radio, LadoLateral, AreaLateral);
52 break;
53 case 'E' :
54 //Solicita los datos de la aplicación
55 printf("\nIngrese la longitud del semieje mayor de la elipse (metros): ");
56 scanf("%lf", &SemiEjeMayor);
57 printf("Ingrese la longitud del semieje menor de la elipse (metros): ");
58 scanf("%lf", &SemiEjeMenor);
59
60 //Obtiene el área de un elipse e imprime el resultado.
61 AreaElipse = NumeroPi * SemiEjeMayor * SemiEjeMenor;
62 printf("El area de una elipse de semiejes %.2lf y %.2lf es %.2lf metros2",
63       SemiEjeMayor, SemiEjeMenor, AreaElipse);
64 break;
65 default :
66     printf("La aplicacion ingresada no es valida.");
67 }
68 }
69 else
70     printf("El numero de iteraciones debe ser mayor a 5 y menor o igual a 20.");
71 }
72
73 double CalcularNumeroPi(int Cant_Iteraciones)
74 {
75     int Cont = 2;
76     double NumeroPi, X;
77
78     //Se inicializa factor X1 con raíz de 2.
79     X = sqrt(2);
80
81     //Ejecuta cada iteración que calcula la aproximación de Pi.
82     do
83     {
84         //Obtiene el factor de Xn.
85         X = sqrt(2 - 2*sqrt(1 - pow(X, 2)/4));
86
87         Cont++;
88     } while (Cont <= Cant_Iteraciones);
89
90     //Obtiene la aproximación del valor de Pi.
91     NumeroPi = pow(2, Cant_Iteraciones) * X;
92
93     //Devuelve el número Pi calculada.
94     return NumeroPi;
95 }
96

```

4. Aplicaciones de la Raíz de Dos

La Raíz de Dos es probablemente el primer número irracional conocido. Geométricamente equivale a la longitud de la diagonal de un cuadrado cuyo lado es igual a la unidad, lo cual se comprueba aplicando el llamado teorema de Pitágoras. Algunos ejemplos de la aplicación de la Raíz de Dos son:

- El cálculo del **Coseno o Seno de 45 grados** (ver figura 13).

$$\text{Coseno } (45^\circ) = \text{Seno } (45^\circ) = \frac{1}{\sqrt{2}}$$

Figura 13: Aproximación del Coseno o Seno de 45 grados

- El cálculo de las dimensiones de un **rectángulo de plata**, rectángulo cuyos lados están en la proporción 1 y Raíz de Dos. Estos rectángulos son también conocidos como rectángulos tipo A4 en referencia al tamaño del papel A4 (ver figura 14).

$$\frac{\text{Lado Menor}}{\text{Lado Mayor}} = \frac{1}{\sqrt{2}}$$

Figura 14: Proporción de los lados de un Rectángulo de Plata

- El cálculo del volumen del **tetraedro regular** (ver figura 15).

$$\text{Volumen del Tetraedro Regular} = \frac{\sqrt{2}}{12} * \text{Arista}^3$$

Figura 15: Fórmula del volumen de un Tetraedro Regular

La Raíz de Dos se puede aproximar utilizando la siguiente sucesión aritmética (ver figura 16).

$$F_n = \frac{F_{n-1} + \frac{2}{F_{n-1}}}{2} \text{ y } F_0 = 1$$

Figura 16: Aproximación de la Raíz de Dos

Sabiendo que, para un valor de n mayor a diez términos, Fn tenderá al valor de la Raíz de Dos.

Se pide desarrollar un programa en Lenguaje C que permita leer el número de iteraciones con las que se calculará la aproximación de la Raíz de Dos (deberá validar que dicho número sea mayor o igual a 10) y después solicite el ingreso de un caracter que indique el tipo de aplicación con la que se trabajará el valor de la Raíz de Dos (C para el cálculo del coseno de 45 grados, R para el cálculo de las dimensiones de un rectángulo de plata o T para el cálculo del volumen de un tetraedro regular). Considerar que en caso se ingrese un caracter diferente deberá indicar un mensaje de error.

En caso haber seleccionado una letra C, deberá imprimir directamente el valor del coseno de 45 grados utilizando la aproximación con la raíz de dos. En caso haber seleccionado una letra R, deberá solicitar el lado menor del rectángulo para calcular e imprimir tanto el lado mayor como el área del rectángulo. Por último, en caso haya seleccionado una letra T deberá pedir la longitud de la arista del tetraedro regular para calcular e imprimir el volumen de la figura.

La solución deberá contar, además del módulo principal, con un módulo para calcular la Raíz de Dos. Este módulo deberá recibir como parámetro un número con la cantidad de iteraciones a utilizar para la aproximación.

Caso de prueba 1

Ingrese el número de iteraciones para calcular la Raíz de Dos: 4
El número de iteraciones debe ser mayor o igual a 10.

Caso de prueba 2

Ingrese el número de iteraciones para calcular la Raíz de Dos: 25
Ingrese el tipo de aplicación de la Raíz de Dos, (Coseno de 45 grados [C], Rectángulo de Plata [R] o Volumen del Tetraedro [T]): X
La aplicación ingresada no es válida.

Caso de prueba 3

Ingrese el número de iteraciones para calcular la Raíz de Dos: 25
Ingrese el tipo de aplicación de la Raíz de Dos, (Coseno de 45 grados [C], Rectángulo de Plata [R] o Volumen del Tetraedro [T]): C
La aproximación del valor del coseno de 45 grados es 0.707107

Caso de prueba 4

Ingrese el número de iteraciones para calcular la Raíz de Dos: 30
Ingrese el tipo de aplicación de la Raíz de Dos, (Coseno de 45 grados [C], Rectángulo de Plata [R] o Volumen del Tetraedro [T]): R
Ingrese la longitud del lado menor de un rectángulo de plata: 21
Los lados del rectángulo de plata son 21.00 y 29.70; y su área es 623.67

Caso de prueba 5

Ingrese el número de iteraciones para calcular la Raíz de Dos: 45
Ingrese el tipo de aplicación de la Raíz de Dos, (Coseno de 45 grados [C], Rectángulo de Plata [R] o Volumen del Tetraedro [T]): T
Ingrese la longitud de la arista del tetraedro regular: 56.5
El volumen de un tetraedro regular de arista 56.50 es 21255.88

Programa 4: Propuesta de solución - Aplicaciones de la raíz de dos

```
1  #include <stdio.h>
2  #include <math.h>
3
4  double CalcularRaizDeDos(int);
5
6  int main ()
7  {
8      char Tipo_Aplicacion;
9      int Cant_Iteraciones;
10     double RaizDeDos, Coseno45, LadoMenor, LadoMayor, Area;
11     double Arista, Volumen;
12
13     //Consulta el número de iteraciones de la aproximación.
14     printf("Ingrese el numero de iteraciones para calcular la Raíz de Dos: ");
15     scanf("%d", &Cant_Iteraciones);
16
17     if ( Cant_Iteraciones >= 10 )
```

```

18 {
19     //Consulta el tipo de objeto a evaluar.
20     printf("Ingresa el tipo de aplicacion de la Raiz de Dos,\n");
21     printf("(Coseno de 45 grados [C], Rectangulo de Plata [R] o Volumen del Tetraedro [T]): ");
22     scanf("\n%c", &Tipo_Aplicacion);
23
24     //Calcula la Razón Aurea con un número dado de iteraciones.
25     RaizDeDos = CalcularRaizDeDos (Cant_Iteraciones);
26
27     switch ( Tipo_Aplicacion )
28     {
29         case 'C' :
30             //Calcula e imprime la aproximación del Coseno de 45 grados.
31             Coseno45 = 1 / RaizDeDos;
32             printf("\nLa aproximacion del valor del coseno de 45 grados es %lf", Coseno45);
33             break;
34         case 'R' :
35             //Solicita los datos de la aplicación
36             printf("\nIngresa la longitud del lado menor de un rectangulo de plata: ");
37             scanf("%lf", &LadoMenor);
38
39             //Calcula el lado mayor y área del rectángulo de plata.
40             LadoMayor = RaizDeDos * LadoMenor;
41             Area = LadoMayor * LadoMenor;
42
43             //Imprime los resultados obtenidos.
44             printf("Los lados del rectangulo de plata son %2lf y %2lf; y su area es %2lf",
45                 LadoMenor, LadoMayor, Area);
46             break;
47         case 'T' :
48             //Solicita los datos de la aplicación
49             printf("\nIngresa la longitud de la arista del tetraedro regular: ");
50             scanf("%lf", &Arista);
51
52             //Calcula el volumen del tetraedro regular.
53             Volumen = RaizDeDos * pow(Arista, 3) / 12;
54
55             //Imprime los resultados obtenidos.
56             printf("El volumen de un tetraedro regular de arista %2lf es %2lf", Arista, Volumen);
57             break;
58         default :
59             printf("La aplicacion ingresada no es valida.");
60     }
61 }
62 else
63     printf("El numero de iteraciones debe ser mayor o igual a 10.");
64 }
65
66 double CalcularRaizDeDos(int Cant_Iteraciones)
67 {
68     int Cont = 1;
69     double RaizDeDos, F;
70
71     //Se inicializa factor F0 con 1.
72     F = 1;
73
74     //Ejecuta cada iteración que calcula la aproximación de la raíz de dos.
75     do
76     {
77         //Obtiene el factor de Fn.
78         F = (F + 2/F)/2;
79
80         Cont++;
81
82     } while (Cont <= Cant_Iteraciones);
83
84     //Obtiene la aproximación del valor de Pi.

```

```
85     RaizDeDos = F;  
86  
87     //Devuelve la raíz de dos.  
88     return RaizDeDos;  
89 }
```

Puede usar cualquier estructura selectiva por lo que el uso de la estructura selectiva múltiple queda a su criterio

No puede usar estructuras iterativas de entrada controlada ni anidadas.