

FUNDAMENTOS DE PROGRAMACIÓN
LABORATORIO 5
PROPUESTAS DE SOLUCIÓN
SEMESTRE ACADÉMICO 2021-2

Horarios: Todos los horarios

Elaborado por Mag. Sergio Ponce

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 las estructuras algorítmicas selectivas anidadas.
- Al finalizar la sesión, el alumno construirá programas usando estructuras algorítmicas selectivas anidadas.

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. Calcular área bajo la curva 1

Una suma de Riemann es una aproximación del área bajo la curva al dividirla en varios rectángulos dentro de un rango $[a,b]$ (ver figura 1). Podemos aproximar el área bajo la curva al dividirla en varios rectángulos del mismo ancho donde la altura de cada rectángulo es igual al valor de la función en el extremo izquierdo de su base.

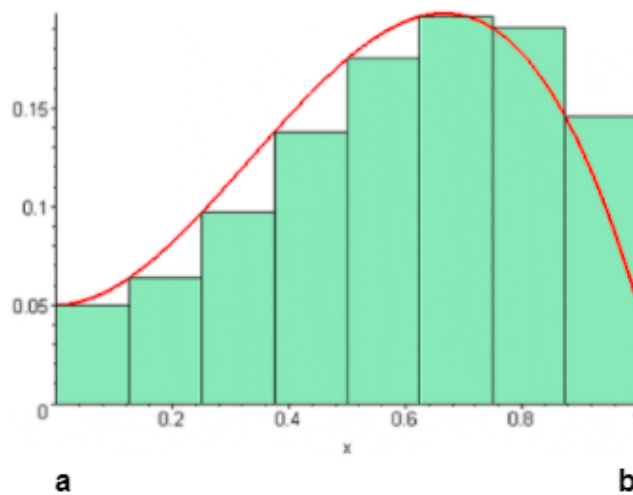


Figura 1: Ilustración de las sumas de Riemann (tomado de: <https://www.lifeder.com/suma-de-riemann/>)

Finalmente para calcular el área utilizando las sumas de Riemann, debe tener en cuenta lo siguiente:

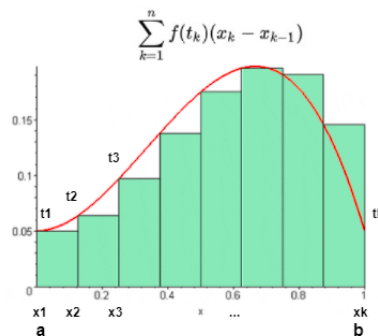


Figura 2: Fórmula para calcular el área en un intervalo [a,b] con n subintervalos utilizando sumas de Riemann (tomado de: <https://www.lifeder.com/suma-de-riemann/>)

La regla o método de Simpson (nombrada así en honor de Thomas Simpson) es un método de integración numérica que se utiliza para obtener la aproximación de una integral en el rango [a,b] (ver figura 3). El método utilizado para la regla de Simpson aproxima los subintervalos mediante polinomios de segundo grado cuya área es la que calculamos y sumamos.

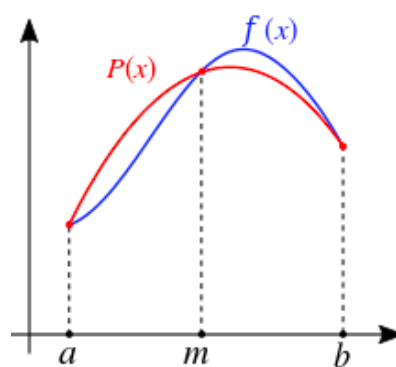


Figura 3: La función $f(x)$ (azul) es aproximada por una función cuadrática $P(x)$ (rojo) (tomado de: https://es.wikipedia.org/wiki/Regla_de_Simpson)

Existen variantes de la regla de Simpson para calcular el área bajo la curva dentro de un intervalo [a,b] y una de

ellas es la regla de Simpson 3/8 compuesta.

Para calcular el área bajo la curva utilizando la regla de Simpson 3/8 compuesta se divide el intervalo $[a,b]$ en n subintervalos iguales (n debe ser múltiplo de 3) de manera que $x_i = a + ih$, donde $h = (b - a)/n$ es la longitud de cada intervalo para $i = 0, 1, \dots, n$

Luego de realizar todo el proceso matemático, la fórmula es la siguiente:

$$\int_a^b f(x)dx = \frac{3h}{8} \left\{ f(a) + 3 \sum_{i=1,4,7,\dots}^{n-2} f(x_i) + 3 \sum_{i=2,5,8,\dots}^{n-1} f(x_i) + 2 \sum_{i=3,6,9,\dots}^{n-2} f(x_i) + f(b) \right\}$$

Figura 4: Fórmula para calcular el área bajo la curva $f(x)$ mediante la regla de Simpson 3/8 compuesta

Se le pide implementar un programa en lenguaje C que permita al usuario ingresar la letra “R” o “T” (también podría ingresar la letra en minúscula) para elegir uno de los métodos para calcular el área bajo una curva de la forma:

$$Ax^3 + Bx^2 + Cx + D$$

- Si ingresa la opción “R” o “r” deberá calcular el área utilizando la Suma de Riemann.
- Si ingresa la opción “T” o “t” deberá calcular el área utilizando el la regla de Simpson 3/8 compuesta.

El programa debe solicitar al usuario ingresar lo siguiente:

- Los coeficientes A, B, C y D.
- El rango del intervalo inferior y superior.
- La cantidad de subintervalos (n) en los que se dividirá el área que deseamos calcular.

El programa debe mostrar mensajes específicos ante las siguientes situaciones:

- Al ingresar la opción debe verificar que sea “R” o “T” (también podría ingresar la letra en minúscula). En caso no se cumpla, se deberá emitir el siguiente mensaje “La opción ingresada no es correcta” y el programa debe terminar.
- Debe validar que el rango inferior del intervalo es menor que el rango superior del intervalo, caso contrario mostrar el mensaje “El valor del límite inferior no puede ser mayor que el superior”.
- Debe validar que la cantidad de subintervalos (n) sea mayor que 0, caso contrario mostrar el mensaje “El valor de n debe ser un número mayor que 0”.
- Para el caso de la regla de Simpson 3/8 compuesta, debe validar que la cantidad de subintervalos (n) sea un número múltiplo de 3, caso contrario mostrar el mensaje “Para utilizar la regla de Simpson 3/8 compuesta n debe ser múltiplo de 3”.

Debe implementar por lo menos los siguientes módulos:

- Un módulo que lea los coeficientes A, B, C y D que el usuario ingrese.
- Un módulo que lea el rango inferior y superior del intervalo y el valor del subintervalo.
- Un módulo que calcule el área mediante la suma de Riemann.

- Un módulo que calcule el área mediante la regla de Simpson 3/8 compuesta.

Caso de prueba 1

Ingrese la opción con la que desea calcular el área de la función:

Suma de Riemann (R o r)

Regla de Simpson 3/8 compuesta (T o t)

p

La opción ingresada no es correcta

Caso de prueba 2

Ingrese la opción con la que desea calcular el área de la función:

Suma de Riemann (R o r)

Regla de Simpson 3/8 compuesta (T o t)

r

Ingrese los coeficientes de la función ($Ax^3 + Bx^2 + Cx + D$):

1 -6 11 -6

Ingrese el límite inferior y superior del intervalo:

3 1

Ingrese la cantidad de subintervalos (n):

10

El valor del límite inferior no puede ser mayor que el superior

Caso de prueba 3

Ingrese la opción con la que desea calcular el área de la función:

Suma de Riemann (R o r)

Regla de Simpson 3/8 compuesta (T o t)

t

Ingrese los coeficientes de la función ($Ax^3 + Bx^2 + Cx + D$):

1 5 -3.4 1

Ingrese el límite inferior y superior del intervalo:

0 10

Ingrese la cantidad de subintervalos (n):

-4

El valor de n debe ser un número mayor que 0

Caso de prueba 4

Ingrese la opción con la que desea calcular el área de la función:

Suma de Riemann (R o r)

Regla de Simpson 3/8 compuesta (T o t)

R

Ingrese los coeficientes de la función ($Ax^3 + Bx^2 + Cx + D$):

1 -6 11 -6

Ingrese el límite inferior y superior del intervalo:

1.3 1.8

Ingrese la cantidad de subintervalos (n):

10000

El área calculando mediante las sumas de Riemann es: 0.165379

Caso de prueba 5

Ingrese la opción con la que desea calcular el área de la función:

Suma de Riemann (R o r)

Regla de Simpson 3/8 compuesta (T o t)

t

Ingrese los coeficientes de la función ($Ax^3 + Bx^2 + Cx + D$):

4 5 0 0

Ingrese el limite inferior y superior del intervalo:

1 5

Ingrese la cantidad de subintervalos (n):

14

Para utilizar la regla de Simpson 3/8 compuesta n debe ser múltiplo de 3.

Caso de prueba 6

Ingrese la opción con la que desea calcular el área de la función:

Suma de Riemann (R o r)

Regla de Simpson 3/8 compuesta (T o t)

t

Ingrese los coeficientes de la función ($Ax^3 + Bx^2 + Cx + D$):

1 5 -3.4 1

Ingrese el limite inferior y superior del intervalo:

0 10

Ingrese la cantidad de subintervalos (n):

-4

El valor de n debe ser un número mayor que 0

Caso de prueba 7

Ingrese la opción con la que desea calcular el área de la función:

Suma de Riemann (R o r)

Regla de Simpson 3/8 compuesta (T o t)

T

Ingrese los coeficientes de la función ($Ax^3 + Bx^2 + Cx + D$):

3 -8.5 7 0

Ingrese el limite inferior y superior del intervalo:

0 2

Ingrese la cantidad de subintervalos (n):

561

El área calculando mediante la regla de Simpson 3/8 compuesta es: 3.333333

Programa 1: Propuesta de solución - Calcular área bajo la curva 1

```
1 #include <stdio.h>
2 #include <math.h>
3
4 void leerCoeficientes(double *,double *,double *,double *);
5 void leerIntervalosyn(double *, double *, int *);
6 double calcularFuncion(double,double,double,double,double);
7 double calcularRiemann(double , double , double ,double ,int , double, double);
8 double hallarTerminosSimpson38(int ,double ,double ,double ,double , double , double , int );
9 double calcularSimpson38(double, double, double ,double ,int , double, double);
10
11 int main() {
12     char opcion;
13     double a,b,c,d,inferior, superior, area;
14     int n;
```

```

15 printf("Ingrese la opción con la que desea calcular el área de la función:\n");
16 printf("Suma de Riemann (R o r)\n");
17 printf("Regla de Simpson 3/8 compuesta (T o t)\n");
18 scanf(" %c", &opcion);
19
20
21 if (opcion=='R' || opcion=='r' || opcion=='T' || opcion=='t' ) {
22     leerCoeficientes(&a,&b,&c,&d);
23     leerIntervalosyn(&inferior,&superior,&n);
24     if (n>0 && inferior<=superior) {
25         if (opcion=='R' || opcion=='r') {
26             /*Riemann*/
27             area=calcularRiemann(a,b,c,d,n,inferior,superior);
28             printf("El área calculando mediante las sumas de Riemann es: %lf", area);
29         } else {
30             /*Simpson 3/8 compuesta*/
31             if (n%3==0) {
32                 area=calcularSimpson38(a,b,c,d,n,inferior,superior);
33                 printf("El área calculando mediante la regla de Simpson 3/8 compuesta es: %lf", area);
34             } else {
35                 printf("Para utilizar la regla de Simpson 3/8 compuesta n debe ser múltiplo de 3.\n");
36             }
37         }
38     } else if (n<=0 && inferior<superior) {
39         printf("El valor de n debe ser un número mayor que 0");
40     } else if (n>0 && inferior>superior) {
41         printf("El valor del límite inferior no puede ser mayor que el superior");
42     }
43 } else {
44     printf("La opción ingresada no es correcta\n");
45 }
46 return 0;
47 }
48
49 void leerCoeficientes(double *a,double *b,double *c,double *d) {
50     printf("Ingrese los coeficientes de la función (Ax3 + Bx2 + Cx+ D):\n");
51     scanf(" %lf %lf %lf %lf",a,b,c,d);
52 }
53 void leerIntervalosyn(double *inferior, double *superior, int *n) {
54     printf("Ingrese el limite inferior y superior del intervalo:\n");
55     scanf(" %lf %lf",inferior, superior);
56     printf("Ingrese la cantidad de subintervalos (n):\n");
57     scanf(" %d",n);
58 }
59 double calcularFuncion(double x,double a, double b, double c, double d) {
60     return (a*pow(x,3))+(b*pow(x,2))+(c*x)+d;
61 }
62 double calcularRiemann(double a,double b,double c,double d,int n, double inferior, double superior) {
63     double i;
64     double areaTotal, area, deltaX,altura;
65     i=inferior;
66     deltaX=(superior-inferior)/n;
67     while (i<=superior) {
68         /*altura*/
69         altura=calcularFuncion(i,a,b,c,d);
70         area=altura*deltaX;
71         areaTotal=areaTotal+area;
72         i=i+deltaX;
73     }
74     return areaTotal;
75 }
76 double hallarTerminosSimpson38(int j,double a,double b,double c,double d, double x, double h, int n) {
77     double sumatoria=0, termino;
78     int maxN;
79     if (j==1) {
80         maxN=n-2;
81     } else if (j==2) {

```

```

82         maxN=n-1;
83     } else if (j==3) {
84         maxN=n-3;
85     }
86     while (j<=maxN) {
87         termino=calcularFuncion(x+j*h,a,b,c,d);
88         sumatoria=sumatoria+termino;
89         j=j+3;
90     }
91     return sumatoria;
92 }
93 double calcularSimpson38(double a,double b,double c,double d,int n, double inferior, double superior) {
94     int j=1;
95     double yInicial, yFinal, h, x, areaTotal, sumatoria1,sumatoria2,sumatoria3;
96     h=(superior-inferior)/n;
97     yInicial=calcularFuncion(inferior,a,b,c,d);
98     yFinal=calcularFuncion(superior,a,b,c,d);
99     x=inferior;
100    sumatoria1=hallarTerminosSimpson38(1,a,b,c,d,x,h,n);
101    sumatoria2=hallarTerminosSimpson38(2,a,b,c,d,x,h,n);
102    sumatoria3=hallarTerminosSimpson38(3,a,b,c,d,x,h,n);
103    areaTotal=(3*h)/8*(yInicial+yFinal+3*sumatoria1+3*sumatoria2+2*sumatoria3);
104    return areaTotal;
105 }

```

2. Calcular área bajo la curva 2

La regla o método de Simpson (nombrada así en honor de Thomas Simpson) es un método de integración numérica que se utiliza para obtener la aproximación de una integral en el rango $[a,b]$. El método utilizado para la regla de Simpson aproxima los subintervalos mediante polinomios de segundo grado cuya área es la que calculamos y sumamos.

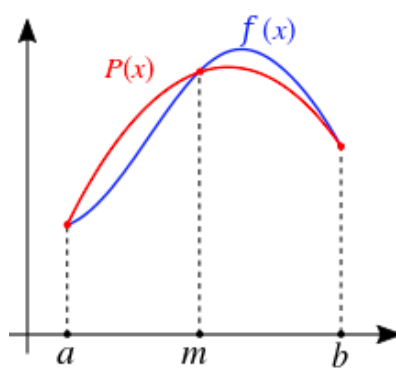


Figura 5: La función $f(x)$ (azul) es aproximada por una función cuadrática $P(x)$ (rojo) (tomado de: https://es.wikipedia.org/wiki/Regla_de_Simpson)

Existen variantes de la regla de Simpson para calcular el área bajo la curva dentro de un intervalo $[a,b]$ y una de ellas es la regla de Simpson 1/3 compuesta.

Para el caso de la Regla de Simpson 1/3 compuesta se divide el intervalo $[a,b]$ en n subintervalos iguales (n debe ser un número par) de manera que $x_j = a + jh$, donde $h = (b - a)/n$ es la longitud de cada intervalo para $j = 0, 1, \dots, n$

Luego de realizar todo el proceso matemático, la fórmula es la siguiente:

$$\int_a^b f(x)dx \approx \frac{h}{3} \left\{ f(a) + 2 \sum_{j=1}^n f(x_{2j}) + 4 \sum_{j=1}^n f(x_{2j-1}) + f(b) \right\}$$

Figura 6: Fórmula para calcular el área bajo la curva $f(x)$ mediante la regla de Simpson 1/3 compuesta

En el método del trapecio compuesto se puede aproximar el área bajo la curva utilizando varios trapecios dentro de un rango $[a,b]$ (ver figura 3), el área bajo la curva se puede aproximar por el de la función lineal que pasa a través de los puntos $(a, f(a))$ y $(b, f(b))$ para cada uno de los trapecios. El área bajo la curva será igual a la suma de las áreas de los trapecios bajo la gráfica.

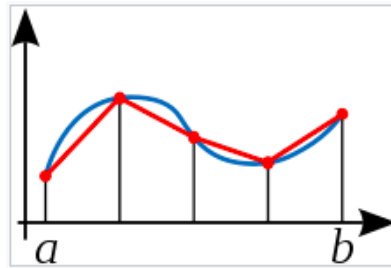


Figura 7: Ilustración del método del trapecio compuesto (tomado de: https://es.wikipedia.org/wiki/Regla_del_trapecio)

Luego de realizar todo el proceso matemático, la fórmula es la siguiente:

$$\int_a^b f(x)dx \sim \frac{b-a}{n} \left[\frac{f(a) + f(b)}{2} + \sum_{k=1}^{n-1} f\left(a + k \frac{b-a}{n}\right) \right]$$

Figura 8: Fórmula para calcular el área en un intervalo $[a,b]$ con n subintervalos utilizando el método del trapecio compuesto (tomado de: https://es.wikipedia.org/wiki/Regla_del_trapecio)

Se le pide implementar un programa en lenguaje C que permita al usuario ingresar la letra “G” o “S” (también podría ingresar la letra en minúscula) para elegir uno de los métodos para calcular el área bajo una curva de la forma:

$$Ax^3 + Bx^2 + Cx + D$$

- Si ingresa la opción “G” o “g” deberá calcular el área utilizando la regla de Simpson 1/3 compuesta.
- Si ingresa la opción “S” o “s” deberá calcular el área utilizando el método del trapecio compuesto.

El programa debe solicitar al usuario ingresar lo siguiente:

- Los coeficientes A, B, C y D.
- El rango del intervalo inferior y superior.
- La cantidad de subintervalos (n) en los que se dividirá el área que deseamos calcular.

El programa debe mostrar mensajes específicos ante las siguientes situaciones:

- Al ingresar la opción debe verificar que sea G” o “S” (también podría ingresar la letra en minúscula). En caso no se cumpla, se deberá emitir el siguiente mensaje “La opción ingresada no es correcta” y el programa debe terminar.
- Debe validar que el rango inferior del intervalo es menor que el rango superior del intervalo, caso contrario mostrar el mensaje “El valor del límite inferior no puede ser mayor que el superior”.
- Debe validar que la cantidad de subintervalos (n) sea mayor que 0, caso contrario mostrar el mensaje “El valor de n debe ser un número mayor que 0”.
- Para el caso de la regla de Simpson 1/3 compuesta, debe validar que la cantidad de subintervalos (n) sea un número par, caso contrario mostrar el mensaje “Para utilizar la regla de Simpson 1/3 compuesta n debe ser par”.

Debe implementar por lo menos los siguientes módulos:

- Un módulo que lea los coeficientes A, B, C y D que el usuario ingrese.
- Un módulo que lea el rango inferior y superior del intervalo y el valor del subintervalo.
- Un módulo que calcule el área mediante la regla de Simpson 1/3 compuesta.
- Un módulo que calcule el área mediante el método del trapecio compuesto.

Caso de prueba 1

Ingrese la opción con la que desea calcular el área de la función:

Regla de Simpson 1/3 compuesta (G o g)

Método del trapecio compuesto (S o s)

p

La opción ingresada no es correcta

Caso de prueba 2

Ingrese la opción con la que desea calcular el área de la función:

Regla de Simpson 1/3 compuesta (G o g)

Método del trapecio compuesto (S o s)

G

Ingrese los coeficientes de la función ($Ax^3 + Bx^2 + Cx + D$):

3 6 -4 0

Ingrese el límite inferior y superior del intervalo:

3 20

Ingrese la cantidad de subintervalos (n):

21

Para utilizar la regla de Simpson 1/3 compuesta, n debe ser múltiplo de 2.

Caso de prueba 3

Ingrese la opción con la que desea calcular el área de la función:

Regla de Simpson 1/3 compuesta (G o g)

Método del trapecio compuesto (S o s)

s

Ingrese los coeficientes de la función ($Ax^3 + Bx^2 + Cx + D$):

1 5 -3.4 1

Ingrese el límite inferior y superior del intervalo:

0 10

Ingrese la cantidad de subintervalos (n):

-4

El valor de n debe ser un número mayor que 0

Caso de prueba 4

Ingrese la opción con la que desea calcular el área de la función:

Regla de Simpson 1/3 compuesta (G o g)

Método del trapecio compuesto (S o s)

G

Ingrese los coeficientes de la función ($Ax^3 + Bx^2 + Cx + D$):

1 -6 11 -6 Ingrese el límite inferior y superior del intervalo:

3 1

Ingrese la cantidad de subintervalos (n):

10

El valor del límite inferior no puede ser mayor que el superior

Caso de prueba 5

Ingrese la opción con la que desea calcular el área de la función:

Regla de Simpson 1/3 compuesta (G o g)

Método del trapecio compuesto (S o s)

S

Ingrese los coeficientes de la función ($Ax^3 + Bx^2 + Cx + D$):

3 -8.5 7 0

Ingrese el límite inferior y superior del intervalo:

0 2

Ingrese la cantidad de subintervalos (n):

999

El área calculando mediante el método del Trapecio compuesto es: 3.325390

Caso de prueba 6

Ingrese la opción con la que desea calcular el área de la función:

Regla de Simpson 1/3 compuesta (G o g)

Método del trapecio compuesto (S o s)

g

Ingrese los coeficientes de la función:

1 -6 11 -6

Ingrese el límite inferior y superior del intervalo:

1.3 1.8

Ingrese la cantidad de subintervalos (n):

10000

El área calculando mediante la regla de Simpson 1/3 compuesta es: 0.165390

Caso de prueba 7

Ingrese la opción con la que desea calcular el área de la función:

Regla de Simpson 1/3 compuesta (G o g)

Método del trapecio compuesto (S o s)

g

Ingrese los coeficientes de la función ($Ax^3 + Bx^2 + Cx + D$):

3 -8.5 7 0

Ingrese el limite inferior y superior del intervalo:

0 2

Ingrese la cantidad de subintervalos (n):

560

El área calculando mediante la regla de Simpson 1/3 compuesta es: 3.328694

Programa 2: Propuesta de solución - Calcular área bajo la curva 2

```
1  #include <stdio.h>
2  #include <math.h>
3
4  void leerCoeficientes(double *,double *,double *,double *);
5  void leerIntervalosyn(double *, double *, int *);
6  double calcularFuncion(double,double,double,double,double);
7  double calcularSimpson13(double , double , double ,double ,int , double, double);
8  double calcularTrapecio(double, double, double ,double ,int , double, double);
9
10 int main() {
11     char opcion;
12     double a,b,c,d,inferior, superior, area;
13     int n;
14
15     printf("Ingrese la opción con la que desea calcular el área de la función:\n");
16     printf("Regla de Simpson 1/3 compuesta (G o g)\n");
17     printf("Método del trapecio compuesto (S o s)\n");
18     scanf(" %c", &opcion);
19
20     if (opcion=='G' || opcion=='g' || opcion=='S' || opcion=='s' ) {
21         leerCoeficientes(&a,&b,&c,&d);
22         leerIntervalosyn(&inferior,&superior,&n);
23         if (n>0 && inferior<=superior) {
24             if (opcion=='G' || opcion=='g') {
25                 /*Simpson 1/3 compuesta*/
26                 if (n %2==0) {
27                     area=calcularSimpson13(a,b,c,d,n,inferior,superior);
28                     printf("El área calculando mediante la regla de Simpson 1/3 compuesta es: %lf", area);
29                 } else {
30                     printf("Para utilizar la regla de Simpson 1/3 compuesta n debe ser múltiplo de 2.\n");
31                 }
32             } else {
33                 /*Trapecio*/
34                 area=calcularTrapecio(a,b,c,d,n,inferior,superior);
35                 printf("El área calculando mediante el metodo del Trapecio compuesto es: %lf", area);
36             }
37         } else if (n<=0 && inferior<superior) {
38             printf("El valor de n debe ser un número mayor que 0");
39         } else if (n>0 && inferior>superior) {
40             printf("El valor del límite inferior no puede ser mayor que el superior");
41         }
42     } else {
43         printf("La opción ingresada no es correcta\n");
44     }
45     return 0;
46 }
47 void leerCoeficientes(double *a,double *b,double *c,double *d) {
48     printf("Ingrese los coeficientes de la función ( $Ax^3 + Bx^2 + Cx + D$ ):\n");
```

```

49     scanf("%lf %lf %lf %lf",a,b,c,d);
50 }
51 void leerIntervalosyn(double *inferior, double *superior, int *n) {
52     printf("Ingrese el limite inferior y superior del intervalo:\n");
53     scanf("%lf %lf",inferior, superior);
54     printf("Ingrese la cantidad de subintervalos (n) que realizara al área que desea calcular\n");
55     scanf("%d",n);
56 }
57 double calcularFuncion(double x,double a, double b, double c, double d) {
58     return (a*pow(x,3))+(b*pow(x,2))+(c*x)+d;
59 }
60 double calcularSimpson13(double a,double b,double c,double d,int n, double inferior, double superior) {
61     double h, yInicial, yFinal, areaTotal,x,termino,sumatoria=0, sumaKimpar=0, sumaKpar=0;
62     int k=1;
63     h=(superior-inferior)/n;
64     yInicial=calcularFuncion(inferior,a,b,c,d);
65     yFinal=calcularFuncion(superior,a,b,c,d);
66     x=inferior+k*h;
67
68     while (k<=n) {
69         termino=calcularFuncion(x,a,b,c,d);
70         if (k %2!=0) {
71             /*posición impar*/
72             sumaKimpar=sumaKimpar+termino;
73         } else {
74             /*posición par*/
75             sumaKpar=sumaKpar+termino;
76         }
77         x=inferior+k*h;
78         k++;
79     }
80     sumatoria= 4*sumaKimpar+2*sumaKpar;
81     areaTotal=(yInicial+yFinal+sumatoria)*(h/3);
82     return areaTotal;
83 }
84
85 double calcularTrapezio(double a,double b,double c,double d,int n, double inferior, double superior) {
86     double h;
87     double yInicial, yFinal, areaTotal,x,termino,sumatoria;
88     int k=1;
89
90     h=(superior-inferior)/n;
91     yInicial=calcularFuncion(inferior,a,b,c,d);
92     yFinal=calcularFuncion(superior,a,b,c,d);
93     x=inferior+k*h;
94     while (k<=n-1) {
95         termino=calcularFuncion(x,a,b,c,d);
96         sumatoria=sumatoria+termino;
97         x=inferior+k*h;
98         k++;
99     }
100     areaTotal=((yInicial+yFinal)/2+sumatoria)*h;
101     return areaTotal;
102 }

```

3. Circunferencias y rectas

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

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

- Solicitar al usuario la cantidad de veces (n) que desea realizar el cálculo (n debe encontrarse entre 1 y 20).
- Por cada iteración deberá ingresar los coeficientes C , D y E de dos circunferencias de la forma $x^2 + y^2 + Cx + Dy + E = 0$.
- Por cada iteración deberá mostrar si las circunferencias se cruzan o no.
- Al finalizar debe mostrar cuantas veces en total se identificaron circunferencias que se cruzan, también debe mostrar para cada cuadrante la cantidad de puntos que son centro de una circunferencia.

Para ello deberá implementar, por lo menos, los siguientes módulos:

- Un módulo que lea los coeficientes C , D y E .
- Un módulo que permita obtener el centro de la circunferencia y el radio.
- Un módulo que permita calcular la distancia entre dos puntos.
- Un módulo que cuente los puntos que se encuentren en un determinado cuadrante.

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

- Solicitar a usuario la cantidad de veces (n) que desea realizar el cálculo (n debe encontrarse entre 1 y 20).
- Por cada iteración deberá ingresar los coeficientes de dos rectas de la forma $Ax + By + C = 0$.
- Por cada iteración deberá mostrar si las rectas son coincidentes, paralelas o secantes.

Deberá implementar, por lo menos, los siguientes módulos:

- Un módulo que lea los coeficientes A , B y C .
- Un módulo que determine e imprima si las rectas son coincidentes, paralelas o secantes.

El programa debe mostrar mensajes específicos ante las siguientes situaciones:

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

Recordar que:

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

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

$$radio_1 + radio_2 \geq distancia$$

donde:

- $radio_1$ es el radio de la circunferencia 1.
 - $radio_2$ es el radio de la circunferencia 2.
 - $distancia$ es la distancia entre los centros de las circunferencias.
- Para calcular el radio (r) y el centro de la circunferencia (h,k) debe tener en cuenta que:
 - $h = -(C/2)$
 - $k = -(D/2)$
 - $r = \sqrt{h^2 + k^2 - E}$
 - La distancia entre dos puntos se calcula mediante:

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

Recordar que:

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

- Son coincidentes si:

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

- Son paralelas si:

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

- Son secantes si:

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

Recordar que:

Para conocer en qué cuadrante del plano cartesiano se encuentra el punto $P(x, y)$ debe trabajar de la siguiente manera:

- Cuadrante 1:

$$x > 0 \text{ y } y > 0$$

- Cuadrante 2:

$$x < 0 \text{ y } y > 0$$

- Cuadrante 3:

$$x < 0 \text{ y } y < 0$$

- Cuadrante 4:

$$x > 0 \text{ y } y < 0$$

También puede apoyarse en la siguiente figura:

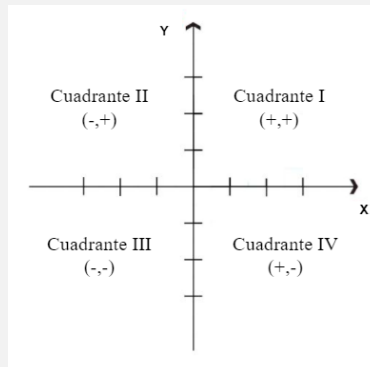


Figura 9: Plano cartesiano

Nota:

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

Caso de prueba 1

Ingrese la opción que desea calcular:

Circunferencias que se intersectan (A o a)

Relación entre rectas (B o b)

R

Error en la opción ingresada.

Caso de prueba 2

Ingrese la opcion que desea calcular:

Circunferencias que se intersectan (A o a)

Relación entre rectas (B o b)

a

Ingrese la cantidad de veces que desea validar si 2 circunferencias de la forma $x^2 + y^2 + Cx + Dy + E$ se cruzan:21

El valor de n no se encuentra dentro del rango solicitado.

Caso de prueba 3

Ingrese la opcion que desea calcular:

Circunferencias que se cruzan (A o a)

Relación entre rectas (B o b)

a

Ingrese la cantidad de veces que desea validar si 2 circunferencias de la forma $x^2 + y^2 + Cx + Dy + E$ se cruzan:1

Ingrese los coeficientes C, D y E de la circunferencia 1 ($x^2 + y^2 + Cx + Dy + E$):2 -8 13

Ingrese los coeficientes C, D y E de la circunferencia 2 ($x^2 + y^2 + Cx + Dy + E$):-40 8 79

Las circunferencias no se cruzan.

Se encontraron 0 cruces entre circunferencias.

En el cuadrante 1 se encontraron 0 puntos que son centro de una circunferencia

En el cuadrante 2 se encontraron 1 puntos que son centro de una circunferencia

En el cuadrante 3 se encontraron 0 puntos que son centro de una circunferencia

En el cuadrante 4 se encontraron 1 puntos que son centro de una circunferencia

Caso de prueba 4

Ingrese la opcion que desea calcular:

Circunferencias que se cruzan (A o a)

Relación entre rectas (B o b)

A

Ingrese la cantidad de veces que desea validar si 2 circunferencias de la forma $x^2 + y^2 + Cx + Dy + E$ se cruzan:3

Ingrese los coeficientes C, D y E de la circunferencia 1 ($x^2 + y^2 + Cx + Dy + E$):10 5 -1

Ingrese los coeficientes C, D y E de la circunferencia 2 ($x^2 + y^2 + Cx + Dy + E$):4 -5 1

Las circunferencias se cruzan

Ingrese los coeficientes C, D y E de la circunferencia 2 ($x^2 + y^2 + Cx + Dy + E$):5 1 -2

Ingrese los coeficientes C, D y E de la circunferencia 3 ($x^2 + y^2 + Cx + Dy + E$):2 -8 7

Las circunferencias se cruzan

Ingrese los coeficientes C, D y E de la circunferencia 3 ($x^2 + y^2 + Cx + Dy + E$):1 2 3

Ingrese los coeficientes C, D y E de la circunferencia 4 ($x^2 + y^2 + Cx + Dy + E$):3 -4 1

Las circunferencias se cruzan

Se encontraron 3 cruces entre circunferencias.

En el cuadrante 1 se encontraron 0 puntos que son centro de una circunferencia

En el cuadrante 2 se encontraron 3 puntos que son centro de una circunferencia

En el cuadrante 3 se encontraron 3 puntos que son centro de una circunferencia

En el cuadrante 4 se encontraron 0 puntos que son centro de una circunferencia

Caso de prueba 5

Ingrese la opcion que desea calcular:

Circunferencias que se intersectan (A o a)

Relación entre rectas (B o b)

b

Ingrese la cantidad de veces que desea validar si 2 rectas son secantes, paralelas o coincidentes: 31

El valor de n no se encuentra dentro del rango solicitado.

Caso de prueba 6

Ingrese la opcion que desea calcular:

Circunferencias que se intersectan (A o a)

Relación entre rectas (B o b)

b

Ingrese la cantidad de veces que desea validar si 2 rectas son secantes, paralelas o coincidentes: 1

Ingrese los coeficientes A, B y C de la recta 1 ($Ax + By + C = 0$): 62.82 62.94 -11202.84

Ingrese los coeficientes A, B y C de la recta 2 ($Ax + By + C = 0$): -23.42 -30.74 7445.18

Las rectas son secantes.

Caso de prueba 7

Ingrese la opcion que desea calcular:

Circunferencias que se intersectan (A o a)

Relación entre rectas (B o b)

b

Ingrese la cantidad de veces que desea validar si 2 rectas son secantes, paralelas o coincidentes: 1

Ingrese los coeficientes A, B y C de la recta 1 ($Ax + By + C = 0$): 67.82 62.94 -30776.65

Ingrese los coeficientes A, B y C de la recta 2 ($Ax + By + C = 0$): 67.82 62.94 -47813.37

Las rectas son paralelas.

Caso de prueba 8

Ingrese la opcion que desea calcular:

Circunferencias que se intersectan (A o a)

Relación entre rectas (B o b)

b

Ingrese la cantidad de veces que desea validar si 2 rectas son secantes, paralelas o coincidentes: 1

Ingrese los coeficientes A, B y C de la recta 1 ($Ax + By + C = 0$): 1 -2 3

Ingrese los coeficientes A, B y C de la recta 2 ($Ax + By + C = 0$): -2 4 -6

Las rectas son coincidentes.

Caso de prueba 9

Ingrese la opcion que desea calcular:

Circunferencias que se intersectan (A o a)

Relación entre rectas (B o b)

b

Ingrese la cantidad de veces que desea validar si 2 rectas son secantes, paralelas o coincidentes: 3

Ingrese los coeficientes A, B y C de la recta 1 ($Ax + By + C = 0$): 2 3 -1

Ingrese los coeficientes A, B y C de la recta 2 ($Ax + By + C = 0$): 4 6 -5

Las rectas son paralelas.

Ingrese los coeficientes A, B y C de la recta 2 ($Ax + By + C = 0$): 1 -2 3

Ingrese los coeficientes A, B y C de la recta 3 ($Ax + By + C = 0$): -2 4 -6

Las rectas son coincidentes.

Ingrese los coeficientes A, B y C de la recta 3 ($Ax + By + C = 0$): 15 6 -1

Ingrese los coeficientes A, B y C de la recta 4 ($Ax + By + C = 0$): 54 -13 6

Las rectas son secantes.

Programa 3: Propuesta de solución - Circunferencias y rectas

```
1  #include <stdio.h>
2  #include <math.h>
3  void leerCoeficientes(double *, double *, double *, int);
4  void obtenerCentroRadio(double, double, double, double *, double *, double *);
5  double calcularDistancia(double, double, double, double);
6  void contarCuadrante(double, double, int *, int *, int *, int *);
7  void leerCoeficientesRecta(double *, double *, double *, int);
8  void relacionEntreRectas(double, double, double, double, double, double);
9
10 int main() {
11     char opcion;
12     int cont=1, n,noCruce=0, cruce=0, cont1=0, cont2=0, cont3=0, cont4=0;
13     double h1,k1,h2,k2,r1,r2,C1,D1,E1,C2,D2,E2,distancia;
14     double A1,B1,A2,B2;
15     printf("Ingrese la opcion que desea calcular:\n");
16     printf("Circunferencias que se cruzan (A o a)\n");
17     printf("Relación entre rectas (B o b)\n");
18     scanf("%c", &opcion);
19
20     if (opcion=='A' || opcion=='a' || opcion=='B' || opcion=='b') {
21         if (opcion=='A' || opcion=='a') {
22             /*Opción 1*/
23             printf("Ingrese la cantidad de veces que desea validar si 2 circunferencias de la forma x^2+y^2+Cx+Dy+E se cruzan:");
24             scanf("%d", &n);
25             if (n>=1 && n<=20) {
26                 while (cont<=n) {
27                     /*solicitar los coeficientes de las ecuaciones de las circunferencias*/
28                     leerCoeficientes(&C1, &D1, &E1,cont);
29                     leerCoeficientes(&C2, &D2, &E2,cont+1);
30                     obtenerCentroRadio(C1,D1,E1,&h1,&k1,&r1);
31                     obtenerCentroRadio(C2,D2,E2,&h2,&k2,&r2);
32                     distancia=calcularDistancia(h1,k1,h2,k2);
33                     contarCuadrante(h1,k1,&cont1,&cont2,&cont3,&cont4);
34                     contarCuadrante(h2,k2,&cont1,&cont2,&cont3,&cont4);
35
36                     if (r1+r2<distancia) {
37                         printf("Las circunferencias no se cruzan.\n");
38                         noCruce++;
39                     } else{
40                         printf("Las circunferencias se cruzan\n");
41                         cruce++;
42                     }
43                     cont++;
44                 }
45             }
46         }
47     }
```

```

45         printf("Se encontraron %d cruces entre circunferencias.\n",cruce);
46         printf("En el cuadrante 1 se encontraron %d puntos que son centro de una circunferencia\n", cont1);
47         printf("En el cuadrante 2 se encontraron %d puntos que son centro de una circunferencia\n", cont2);
48         printf("En el cuadrante 3 se encontraron %d puntos que son centro de una circunferencia\n", cont3);
49         printf("En el cuadrante 4 se encontraron %d puntos que son centro de una circunferencia\n", cont4);

50     } else {
51         printf("El valor de n no se encuentra dentro del rango solicitado.");
52     }
53 } else {
54     /*opcion 2*/
55     printf("Ingrese la cantidad de veces que desea validar si 2 rectas son secantes, paralelas o coincidentes
56           :");
57     scanf("%d", &n);
58     if (n>=1 && n<=20) {
59         while (cont<=n) {
60             /*solicitar ingresar los coeficientes de las rectas*/
61             leerCoeficientesRecta(&A1,&B1,&C1,cont);
62             leerCoeficientesRecta(&A2,&B2,&C2,cont+1);
63             relacionEntreRectas(A1, B1, C1, A2, B2, C2);
64             cont++;
65         }
66     } else {
67         printf("El valor de n no se encuentra dentro del rango solicitado.");
68     }
69 } else {
70     printf("Error en la opción ingresada.\n");
71 }
72 return 0;
73 }
74 void leerCoeficientes(double *C, double *D, double *E, int i) {
75     printf("Ingrese los coeficientes C, D y E de la circunferencia %d ( $x^2+y^2+Cx+Dy+E=0$ ):", i);
76     scanf("%lf %lf %lf",C, D, E);
77 }
78 void obtenerCentroyRadio(double C, double D, double E,double *h,double *k,double *r) {
79     *h=-C/2;
80     *k=-D/2;
81     *r=sqrt(pow(*h,2)+pow(*k,2)-E);
82 }
83 double calcularDistancia(double h1 ,double k1 ,double h2, double k2) {
84     return sqrt(pow(h2-h1,2)+pow(k2-k1,2));
85 }
86 void contarCuadrante(double h,double k, int *cont1, int *cont2, int *cont3, int *cont4) {
87     int cuadrante1, cuadrante2, cuadrante3, cuadrante4;
88
89     cuadrante1=(h>0 && k>0);
90     cuadrante2=(h<0 && k>0);
91     cuadrante3=(h<0 && k<0);
92     cuadrante4=(h>0 && k<0);
93
94     if (cuadrante1) {
95         (*cont1)++;
96     } else if (cuadrante2) {
97         (*cont2)++;
98     } else if (cuadrante3) {
99         (*cont3)++;
100     } else if (cuadrante4) {
101         (*cont4)++;
102     }
103 }
104 void leerCoeficientesRecta(double *A,double *B,double *C,int i) {
105     printf("Ingrese los coeficientes A, B y C de la recta %d ( $Ax + By + C = 0$ ):", i);
106     scanf("%lf %lf %lf",A, B, C);
107 }
108 void relacionEntreRectas(double A1, double B1, double C1, double A2, double B2, double C2) {
109     double div1,div2,div3, diferencia1, diferencia2;

```

```

110     div1=A1/A2;
111     div2=B1/B2;
112     div3=C1/C2;
113     printf("div1: %lf\n",div1);
114     printf("div2: %lf\n",div2);
115     printf("div3: %lf\n",div3);
116
117     diferencia1=fabs(div1-div2);
118     diferencia2=fabs(div2-div3);
119     if (diferencia1<0.01 && diferencia2<0.01) {
120         printf("Las rectas son coincidentes.\n");
121     } else if (diferencia1<0.01 && !diferencia2<0.01) {
122         printf("Las rectas son paralelas.\n");
123     } else {
124         printf("Las rectas son secantes.\n");
125     }
126 }

```

4. Sumatorias

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

Si ingresa la opción “A” o “a” debe realizar el siguiente cálculo:

$$S = \sum_{j=0}^n \left(\left(\frac{\pi a}{2} \right)^{2j} \frac{a^j b^{j+1}}{(a-b)(2j+1)!} + 4b^{2j+1} \frac{a}{(a-b)(2j)!} \right)$$

Figura 10: Sumatoria 1

Donde:

- La cantidad de términos está representada por la n cuyo rango es $1 \leq n \leq 10$.
- a es un número entre 2 y 6.
- b es un número entre 1 y 5
- a es mayor que b
- S es el resultado de sumar los n términos.

Si ingresa la opción “B” o “b” debe realizar el siguiente cálculo:

$$S = \sum_{j=0}^n \left(\frac{(2j)!}{4^j (j!)^2 (2j+1)} a^{2j+1} \right)$$

Figura 11: Sumatoria 2

Donde:

- La cantidad de términos está representada por la n cuyo rango es $1 \leq n \leq 20$.

- a es un número entre 1 y 3.
- S es el resultado de sumar los n términos.

El programa debe mostrar mensajes específicos ante las siguientes situaciones:

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

Debe implementar por lo menos los siguientes módulos:

- Un módulo que solicite ingresar los datos para las variables a , b y n para el caso de la sumatoria 1.
- Un módulo que solicite ingresar los datos para las variables a y n para el caso de la sumatoria 2.
- Un módulo que calcule la sumatoria 1.
- Un módulo que calcule la sumatoria 2.

Caso de prueba 1

Ingrese la sumatoria que desea calcular:
 Sumatoria 1 (A o a)
 Sumatoria 2 (B o b)
 s
 Error en la opción ingresada.

Caso de prueba 2

Ingrese la sumatoria que desea calcular:
 Sumatoria 1 (A o a)
 Sumatoria 2 (B o b)
 A
 Ingrese el valor n:
 15
 La cantidad de términos debe estar entre 1 y 10.

Caso de prueba 3

Ingrese la sumatoria que desea calcular:
 Sumatoria 1 (A o a)
 Sumatoria 2 (B o b)
 A
 Ingrese el valor n:
 7
 Ingrese los valores de a y b:
 1 4
 El valor de a no se encuentra en el rango solicitado y no es mayor que b.

Caso de prueba 4

Ingrese la sumatoria que desea calcular:

Sumatoria 1 (A o a)

Sumatoria 2 (B o b)

a

Ingrese el valor n:

8

Ingrese los valores de a y b:

2 6

El valor de b no se encuentra en el rango solicitado y a no es mayor que b.

Caso de prueba 5

Ingrese la sumatoria que desea calcular:

Sumatoria 1 (A o a)

Sumatoria 2 (B o b) a

Ingrese el valor n:

5

Ingrese los valores de a y b:

7 2

el valor de a no se encuentra dentro del rango solicitado.

Caso de prueba 6

Ingrese la sumatoria que desea calcular:

Sumatoria 1 (A o a)

Sumatoria 2 (B o b)

A

Ingrese el valor n:

5

Ingrese los valores de a y b:

3 4

el valor de a debe ser mayor que el de b.

Caso de prueba 7

Ingrese la sumatoria que desea calcular:

Sumatoria 1 (A o a)

Sumatoria 2 (B o b)

b

Ingrese los valores de a y n:

2 21

La cantidad de términos debe estar entre 1 y 20.

Caso de prueba 8

Ingrese la sumatoria que desea calcular:

Sumatoria 1 (A o a)

Sumatoria 2 (B o b)

b

Ingrese los valores de a y n:

30 30

El valor de a y n no se encuentran dentro del rango solicitado.

Caso de prueba 9

Ingrese la sumatoria que desea calcular:

Sumatoria 1 (A o a)

Sumatoria 2 (B o b)

B

Ingrese los valores de a y n:

4 19

El valor de a no se encuentra en el rango solicitado.

Caso de prueba 10

Ingrese la sumatoria que desea calcular:

Sumatoria 1 (A o a)

Sumatoria 2 (B o b)

A

Ingrese el valor n:

5

Ingrese los valores de a y b:

2 1

El resultado de la sumatoria "A" es: 24.042614

Caso de prueba 11

Ingrese la sumatoria que desea calcular:

Sumatoria 1 (A o a)

Sumatoria 2 (B o b)

a

Ingrese el valor n:

7

Ingrese los valores de a y b:

3 1

El resultado de la sumatoria "a" es: 1584.123262

Caso de prueba 12

Ingrese la sumatoria que desea calcular:

Sumatoria 1 (A o a)

Sumatoria 2 (B o b)

b

Ingrese los valores de a y n:

2 10

El resultado de la sumatoria "b" es: 221.832707

Programa 4: Propuesta de solución - Sumatorias

```
1 #include <stdio.h>
2 #include <math.h>
3 #define PI 3.1416
4
5 void leerDatosSumatoria1(double *,double *, double *);
6 void leerDatosSumatoria2(double *, int *);
7 double calcularSumatoria1(double , double , int );
8 double calcularSumatoria2(double ,int );
9
10 int main() {
11     char opcion;
12     double a,b,c,d,sumatoria;
```

```

13 int n, validarA, validarB, validarAB, validarN;
14
15 printf("Ingrese la sumatoria que desea calcular:\n");
16 printf("Sumatoria 1 (A o a)\n");
17 printf("Sumatoria 2 (B o b)\n");
18 scanf(" %c", &opcion);
19
20 if (opcion=='A' || opcion=='a' || opcion=='B' || opcion=='b' ) {
21     if (opcion=='A' || opcion=='a') {
22         /*Sumatoria 1*/
23         leerDatosSumatoria1(&a,&b,&n);
24         if (n>=1 && n<=10) {
25             /*a entre 4 y 10 b entre 2 y 8 con a>b*/
26             validarA=(a>=2 && a<=6);
27             validarB=(b>=1 && b<=5);
28             validarAB=(a>b);
29             if (validarA && validarB && validarAB) {
30                 //resolver sumatoria
31                 sumatoria=calcularSumatoria1(a,b,n);
32                 printf("El resultado de la sumatoria \" %c\" es: %lf\n",opcion, sumatoria);
33             } else if (!validarA && validarB && validarAB) {
34                 /*a no está en el rango*/
35                 printf("el valor de a no se encuentra dentro del rango solicitado.\n");
36             } else if (validarA && !validarB && validarAB) {
37                 /*b no está en el rango*/
38                 printf("el valor de b no se encuentra dentro del rango solicitado.\n");
39             } else if (validarA && validarB && !validarAB) {
40                 /*a no es mayor que b*/
41                 printf("el valor de a debe ser mayor que el de b.\n");
42             } else if (!validarA && !validarB && validarAB) {
43                 /*a no está en el rango y b no está en el rango*/
44                 printf("Los valores de a y b no se encuentran en el rango solicitado.\n");
45             } else if (!validarA && validarB && !validarAB) {
46                 /*a no está en el rango y a no es mayor que b*/
47                 printf("El valor de a no se encuentra en el rango solicitado y no es mayor que b.\n"
48                     );
49             } else if (validarA && !validarB && !validarAB) {
50                 /*b no está en el rango y a no es mayor que b*/
51                 printf("El valor de b no se encuentra en el rango solicitado y a no es mayor que b.\n"
52                     );
53             } else {
54                 /*ninguno cumple*/
55                 printf("Lo valores no se encuentran dentro del rango solicitado.\n");
56             }
57         } else {
58             /*Sumatoria 2*/
59             leerDatosSumatoria2(&a,&n);
60             validarN=(n>=1 && n<=20);
61             validarA=(a>=1 && a<=3);
62             if ( validarN && validarA) {
63                 /*Calcular sumatoria*/
64                 sumatoria=calcularSumatoria2(a,n);
65                 printf("El resultado de la sumatoria \" %c\" es: %lf\n",opcion, sumatoria);
66             } else if (!validarA && validarN) {
67                 printf("El valor de a no se encuentra en el rango solicitado.\n");
68             } else if (validarA && !validarN) {
69                 printf("La cantidad de términos debe estar entre 1 y 20.\n");
70             } else {
71                 /*ninguno caso cumple*/
72                 printf("El valor de a y n no se encuentran dentro del rango solicitado.\n");
73             }
74         }
75     } else {
76         printf("Error en la opción ingresada.\n");
77     }

```



```

78     }
79     return 0;
80 }
81 void leerDatosSumatoria1(double *a, double *b, double *n) {
82     printf("Ingrese el valor n:\n");
83     scanf("%d", n);
84     printf("Ingrese los valores de a y b:\n");
85     scanf("%lf %lf", a, b);
86
87
88 }
89 void leerDatosSumatoria2(double *a, int *n) {
90     printf("Ingrese los valores de a y n:\n");
91     scanf("%lf %d", a, n);
92 }
93 double calcularSumatoria1(double a, double b, int n) {
94     int j=0, factorialPares=1, factorialImpares=1;
95     double termino1, termino2, sumatoria=0, numerador1, denominador1, numerador2, denominador2;
96     while (j<=n) {
97
98         if (j==0) {
99             numerador1=a*b*PI;
100             denominador1=2*(a-b);
101             numerador2=4*b*a;
102             denominador2=a-b;
103         } else {
104             numerador1=pow(PI*a, 2*j)*pow(a, j)*pow(b, j+1);
105             factorialImpares=(2*j+1)*(2*j)*factorialImpares;
106             denominador1=pow(2, 2*j)*(a-b)*factorialImpares;
107             numerador2=4*pow(b, 2*j+1)*a;
108             factorialPares=(2*j)*(2*j-1)*factorialPares;
109             denominador2=(a-b)*factorialPares;
110         }
111         termino1=numerador1/denominador1;
112         termino2=numerador2/denominador2;
113         sumatoria=sumatoria+termino1+termino2;
114         j++;
115     }
116     return sumatoria;
117 }
118 double calcularSumatoria2(double a, int n) {
119     int j=1, factorial=1, factorialPares=1;
120     double termino, numerador=a, denominador=1, sumatoria=a;
121     while (j<=n) {
122         factorial=j*factorial;
123         factorialPares=(2*j)*(2*j-1)*factorialPares;
124         numerador=factorialPares*pow(a, 2*j+1);
125
126         denominador=pow(4, j)*pow(factorial, 2)*(2*j+1);
127         termino=numerador/denominador;
128         printf("sumatoria 1: %lf\n", sumatoria);
129         sumatoria=sumatoria+termino;
130         j++;
131     }
132     return sumatoria;
133 }

```

No podrá usar estructuras algorítmicas iterativas anidadas, selectivas múltiples, iterativas de salida controlada ni la instrucción for.