

FUNDAMENTOS DE PROGRAMACIÓN
LABORATORIO 5
PROPUESTAS DE SOLUCIÓN
SEMESTRE ACADÉMICO 2024-0

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 construirá programas usando programación modular usando paso de parámetro por valor y simulando el paso de parámetros por referencia

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. Sumatorias

Se le pide implementar un programa en lenguaje C que muestre un menú de opciones al usuario para que pueda elegir una de ellas y calcular una determinada sumatoria (considerar hasta 10 decimales en su impresión):

Si el usuario ingresa la opción A debe realizar el siguiente cálculo:

$$\sum_{i=1}^n \frac{B_{2i}(-4)^i(1-4^i)}{(2i)!}$$

Figura 1: Sumatoria A

Donde:

- La cantidad máxima de términos está representada por n cuyo rango se encuentra entre $[1,5]$.
- B_{2i} es el número de Bernoulli en la posición $2i$ con $i=1,2,3,4,5$

Si el usuario ingresa la opción **B** debe realizar el siguiente cálculo:

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

Figura 2: Sumatoria B

Donde:

- La cantidad máxima de términos está representada por n cuyo rango se encuentra entre $[1,3]$.
- Considerar el rango de a $[3,5]$.

El programa debe mostrar mensajes específicos ante las siguientes situaciones:

- Al ingresar la opción debe verificar que sea **A** o **B**. En caso no se cumpla, se deberá emitir el siguiente mensaje “La opción ingresada no es válida.” y el programa debe terminar.
- Debe validar que el usuario ingrese datos correctos dependiendo de la sumatoria escogida, caso contrario mostrar el respectivo mensaje de error (ver casos de prueba) y el programa debe terminar.

Debe implementar por lo menos los siguientes módulos:

- Un módulo que calcule la sumatoria **A**.
- Un módulo que calcule la sumatoria **B**.
- Un módulo que retorne el número de Bernoulli en la posición indicada.

Nota:

- Considerar el valor de **PI** = 3.141592
- Considerar los siguientes números de Bernoulli:
 - $B_2 = 0.166666667$
 - $B_4 = -0.033333333$
 - $B_6 = 0.023809524$
 - $B_8 = -0.033333333$
 - $B_{10} = 0.075757576$

Casos de prueba:

```
Ingrese la sumatoria que desea calcular:
-Sumatoria A
-Sumatoria B
Z
La opción ingresada no es válida
```

```
Ingrese la sumatoria que desea calcular:
-Sumatoria A
-Sumatoria B
A
Ingrese el valor de n: -4
El valor de n no es válido
```

```
Ingrese la sumatoria que desea calcular:
-Sumatoria A
-Sumatoria B
A
Ingrese el valor de n: 1
El resultado de la sumatoria A es 1.0000000020
```

```
Ingrese la sumatoria que desea calcular:
-Sumatoria A
-Sumatoria B
A
Ingrese el valor de n: 4
El resultado de la sumatoria A es 1.5206349198
```

```
Ingrese la sumatoria que desea calcular:
-Sumatoria A
-Sumatoria B
B
Ingrese el valor de n: 10
Ingrese el valor de a: 4
Alguno de los valores ingresados no es válido
```

```
Ingrese la sumatoria que desea calcular:
-Sumatoria A
-Sumatoria B
B
Ingrese el valor de n: 1
Ingrese el valor de a: 3
El resultado de la sumatoria B es 21.0295053667
```

```
Ingrese la sumatoria que desea calcular:
-Sumatoria A
-Sumatoria B
B
Ingrese el valor de n: 3
Ingrese el valor de a: 5
El resultado de la sumatoria B es 380.9681758709
```

Para el desarrollo del programa debe usar estructuras iterativas con entrada controlada por contador

Programa 1: Propuesta de solución - Sumatorias

```
1 #include <stdio.h>
2 #include <math.h>
3 #define B_2 0.166666667
4 #define B_4 -0.033333333
5 #define B_6 0.023809524
6 #define B_8 -0.033333333
7 #define B_10 0.075757576
8 #define PI 3.141592
9
```

```

10 double calcularSumatoriaA(int n);
11 double calcularSumatoriaB(int a, int n);
12 double obtenerNumBernoulli(int n);
13
14 int main() {
15     char opcion;
16     int nValido, aValido, n, a, opcionValida, imprimirSumatoria=1;
17     double sumatoria;
18
19     printf("Ingrese la sumatoria que desea calcular:\n");
20     printf("-Sumatoria A\n");
21     printf("-Sumatoria B\n");
22     scanf("%c", &opcion);
23     opcionValida=(opcion=='A') || (opcion=='B');
24     if (opcionValida) {
25         if (opcion=='A') {
26             printf("Ingrese el valor de n: ");
27             scanf("%d",&n);
28             nValido=n>=1 && n<=5;
29             if (nValido) {
30                 sumatoria=calcularSumatoriaA(n);
31             } else {
32                 printf("El valor de n no es válido");
33                 imprimirSumatoria=0;
34             }
35         } else { //opción B
36             printf("Ingrese el valor de n: ");
37             scanf("%d",&n);
38             printf("Ingrese el valor de a: ");
39             scanf("%d",&a);
40             nValido=n>=1 && n<=3;
41             aValido=a>=3 && a<=5;
42             if (nValido && aValido) {
43                 sumatoria=calcularSumatoriaB(n,a);
44             } else {
45                 printf("Alguno de los valores ingresados no es válido");
46                 imprimirSumatoria=0;
47             }
48         }
49         if (imprimirSumatoria) {
50             printf("El resultado de la sumatoria %c es %.10lf",opcion,sumatoria);
51         }
52     } else {
53         printf("La opción ingresada no es válida");
54     }
55 }
56 double calcularSumatoriaA(int n){
57     int i=1, factorialPares=1;
58     double numerador, termino, sumatoria=0, numBernoulli;
59     while (i<=n) {
60         /*No es necesario validar el caso del factorial, pues i empieza en 1*/
61         factorialPares=(2*i)*(2*i-1)*factorialPares;
62         /*el número de Bernoulli cambiará según el valor de i*/
63         numBernoulli=obtenerNumBernoulli(i);
64         numerador=numBernoulli*pow(-4,i)*(1-pow(4,i));
65         termino=numerador/factorialPares;
66         sumatoria=sumatoria+termino;
67         i++;
68     }
69     return sumatoria;
70 }
71 double calcularSumatoriaB(int n, int a) {
72     int i=1, factorialImpares=1;
73     double termino, numerador, denominador, sumatoria=0;
74     while (i<=n) {
75         /*cuando i=0 las variables para calcular los factoriales no deben actualizarse*/
76         if (i>0) {

```

```

77         factorialImpares=(2*i+1)*(2*i)*factorialImpares;
78     }
79     numerador=factorialImpares*pow(a,PI);
80     denominador=pow(3*a,i);
81     termino=numerador/denominador;
82     sumatoria=sumatoria+termino;
83     i++;
84 }
85 return sumatoria;
86 }
87 double obtenerNumBernoulli(int num) {
88     /*num solo puede ser 1,2,3,4,5*/
89     int indice=2*num;
90     double numBernoulli;
91     if (indice==2) {
92         numBernoulli=B_2;
93     } else if (indice==4) {
94         numBernoulli=B_4;
95     } else if (indice==6) {
96         numBernoulli=B_6;
97     } else if (indice==8) {
98         numBernoulli=B_8;
99     } else {
100         numBernoulli=B_10;
101     }
102     return numBernoulli;
103 }

```

No debe usar estructuras algorítmicas iterativas anidadas, iterativas con salida controlada o por centinela.