

Computación Científica Avanzada
Tarea 3
Programación en C
Fecha de entrega: Jueves 22 de Septiembre, 10am

Esta tarea no se debe entregar, no se debe enviar vía email. Los códigos deben estar en un repositorio en github que se llame Tarea3. **La fecha de última actualización no debe ser posterior al 22 de septiembre a las 10am.**

Envíe por correo el link del repositorio **antes del jueves 22 de septiembre a las 10am.**

Los programas se deben poder compilar con un Makefile. Un mismo makefile debe servir para compilar todos los códigos juntos o uno a la vez (esto es evaluable).

Programas de calentamiento

El objetivo de estos programas es permitirle usar el lenguaje de programación. Todos son programas muy sencillos, pero se espera que se desarrollen teniendo en cuenta todas las observaciones que hasta el momento se ha considerado en el curso

- a) Optimización
- b) Desempeño
- c) Un mínimo de diseño y modularidad

1. Escriba un programa que utilice la serie de Taylor de la función seno (expansión al rededor de $x=0$) para calcular el seno de un ángulo. El programa debe recibir como entrada el valor del ángulo (x) en grados y el numero de términos (N) de la serie a usarse en la expansión, y retornar en pantalla el valor de $\text{sen}(x)$.

2. Escriba un programa que utilice la serie de Taylor de la función coseno (expansión al rededor de $x=0$) para calcular el coseno de un ángulo. El programa debe recibir como entrada el valor del ángulo (x) en grados y el numero de términos (N) de la serie a usarse en la expansión, y retornar en pantalla el valor de $\text{cos}(x)$.

3. División entera por dos es aquella en la que $a/2 = b + r$, donde b es el resultado de la división y r es el residuo. Por ejemplo, $17/2 = 15 + 2$, el resultado de la división es 15 y el residuo es 2).

Escriba un programa que calcule el numero de veces que se puede dividir un numero entre dos, asumiendo división entera. (ejemplo, si $9/2=4$, $4/2=2$, $2/2=1$, entonces 9 se puede dividir por dos tres veces.). El programa debe recibir el numero a dividir y debe imprimir en pantalla el numero de veces que dicho numero es divisible (división entera) por dos.

4. Escriba un programa que calcule el factorial de un numero x .

5. Calculadora:

Escriba un programa que calcule la *suma*, *resta*, *multiplicación* y *división* de dos operandos. Para cada operación (*suma*, *resta*, *multiplicación* o *división*) debe escribir una función. El programa debe recibir, en orden, el primer operando, el operador y el tercer operando (ej. *1 suma 2*) y producir en pantalla el resultado de la operación.

6. Escriba un programa que evalúe (en una rutina diferente al main) la función $f(x)=\sin(x)*\ln(2*x+1)*\sinh(x)$ en el intervalo $[0, 10]$. El programa debe recibir el tamaño del paso (dx) entre evaluaciones sucesivas de la función.

El resultado se debe escribir en un archivo de tres columnas. La primer columna conteniendo el valor i del incremento en x ($x=i*dx$), la segunda columna debe contener x , y la tercera conteniendo el valor de $f(x)$.

En otra rutina, escriba un procedimiento que lea la tabla producida en la parte anterior, y escriba otro archivo de dos columnas, en este caso la primer columna debe tener el valor $f(x)$ y la segunda columna el valor x .

El resultado final del programa deben ser los dos archivos. Grafique los resultados usando gnuplot (o el graficador de su preferencia) para verificar la calidad del resultado.

7. Precedencia de operadores.

Escriba un programa que evalúe la siguiente operación

$$\frac{\sin(x) + \cos(x)}{2} (a + b)$$

Utilizando explícitamente la siguiente implementación de la función (implementar cada caso en una función diferente en el mismo programa)

$$f1(x) = \sin(x) + \cos(x)/2 * a + b$$

$$f2(x) = \sin(x) + \cos(x)/2 * (a + b)$$

$$f3(x) = (\sin(x) + \cos(x)/2) * a + b$$

$$f4(x) = ((\sin(x) + \cos(x))/2)*(a+b)$$

- Cual es la implementación correcta?
- En cada caso, en que se diferencia cada implementación de la formula correcta?
- Que aprendió sobre la precedencia de los operadores en este ejemplo?

8. Escriba un programa que calcule los números primos menores que X_{max} . El programa debe recibir X_{max} y como resultado debe imprimir en pantalla todos los números primos menores que X_{max} .

9. Una partícula tiene una masa en reposo (cuando no esta en movimiento) m_0 . Si la velocidad de movimiento de la partícula es muy grande su masa esta dada por

$$m = \frac{m_0}{\sqrt{1 - v^2/c^2}} \quad (*)$$

Cuando un objeto esta en reposo tiene una masa de $1k$. Escriba un programa que calcule el valor de la

masa de dicho objeto para diferentes valores de la velocidad de la partícula. El programa debe recibir el valor del incremento dv entre valores sucesivos de la velocidad y producir un archivo de columnas. La primera columna debe contener el valor de la velocidad de la partícula v , la segunda el valor de v/c y la tercera debe contener el valor de la masa de la partícula.

El programa debe tener una función en la que se evalúa la masa de la partícula (eq. *).

Grafique sus resultados usando gnuplot.

10. Escriba un programa que calcule el número p usando dos aproximaciones diferentes

$$pi1 \rightarrow \frac{2}{\pi} = \frac{\sqrt{2}}{2} \cdot \frac{\sqrt{2+\sqrt{2}}}{2} \cdot \frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2} \dots$$

$$pi2 \rightarrow \frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \dots$$

Calcule p en cada aproximación implementándola en una función distinta en cada caso (en el mismo programa). El programa debe recibir el número de términos en las expansiones a usarse para el cálculo de p y debe imprimir en pantalla los dos valores de p obtenidos.

Programas específicos para medir los efectos de la implementación.

Los siguientes problemas/programas son más orientados a medir los efectos de las observaciones en implementación (HPC1).

11. Se mencionó en clase que hacer un buen uso de los registros puede ser conveniente para efectos de optimizar el uso del tiempo de cómputo.

Escriba un programa que a través de una secuencia de iteraciones muestre que en efecto hacer la optimización

$$t1 = \sin(\alpha) * x + \cos(\alpha) * y;$$

$$t2 = -\cos(\alpha) * x + \sin(\alpha) * y;$$

$$s = \sin(\alpha); c = \cos(\alpha);$$

$$t1 = s * x + c * y;$$

$$t2 = -c * x + s * y$$

conduce a una ganancia de tiempo de CPU.

Haga una gráfica de tiempo de ejecución como función del número de iteraciones y úsela para concluir para cuántas iteraciones se vuelve importante considerar la optimización mencionada

12. Se tiene la siguiente secuencia de operaciones

En el primer caso

```
for i=1,length  
a[i] = b[i] * c
```

En el segundo caso

```
for k=1,nvectors  
for i=1,length  
a[i] = b[i] * c[i]
```

Donde todos los valores de $c[i]$ tienen el mismo valor. Cual código tarda mas?

13. Puede notar alguna diferencia en el tiempo de ejecución del anterior programa si se corre en int, float, double y long double precisión?

14. Escriba un programa que implemente la realización de un loop como el que se presenta a continuación

```
for (loop=0; loop<10; loop++) {  
  for (i=0; i<N; i++) {  
    ... = ... x[i] ...  
  }  
}
```

y luego el mismo código pero usando el segundo orden

```
for (i=0; i<N; i++) {  
  for (loop=0; loop<10; loop++) {  
    ... = ... x[i] ...  
  }  
}
```

Cual ejecución demora mas? Por que?

15. Escriba tres programas para calcular la multiplicación de dos matrices usando como algoritmos centrales las siguientes formas

a) Caso a

```
for i=1..n  
for j=1..n  
for k=1..n
```

$c[i,j] += a[i,k]*b[k,j]$

b) Caso b

```
for i=1..n
for k=1..n
for j=1..n
c[i,j] += a[i,k]*b[k,j]
```

c) Caso c

```
for k=1..n:
for i=1..n:
for j=1..n:
c[i,j] += a[i,k]*b[k,j]
```

Cual método funciona mas rápido? Cambian los resultados con cada implementación?

16. Suponga que se tiene que calcular una sumatoria como la siguiente (de una serie de Taylor de la función $\sin(x)$, por ejemplo)

```
for (i=0; i<N; i++)
s += a[i]*b[i]
```

haga una gráfica del tiempo de ejecución de la misma sumatoria, fracturándola en sumatorias de términos cada 2, 3, 4, 5, 6 y 10 términos.

Hasta donde se observa ganancia en el tiempo de ejecución?