

PRINCIPIO DE COMPUTADORES.

PRÁCTICA 3. CÁLCULO DE SERIES NUMÉRICAS.

Descripción.

En esta práctica vamos a calcular el valor de una serie numérica para aproximarnos a una función. Para ello, utilizaremos el ejemplo concreto de una serie que se aproxima a la función

$$\frac{1}{(1+x)}$$

El sumatorio $\sum_{n=0}^{\infty} (-1)^n x^n$ se aproxima a $\frac{1}{(1+x)}$ siempre y cuando $|x| < 1$

Cuanto más términos del sumatorio elijas, menor será el error que cometes.

Por ejemplo, para $x = 0.6$ (cumple con el requisito de que su valor absoluto es menor que 1)

$1 / (1 + 0.6) = 0.625$ Este sería el valor real del cociente

Si ahora vamos calculando su aproximación con el sumatorio, añadiendo términos, podemos ir viendo el error que se comete en cada iteración:

iteración n = 0

$(-1)^0 (0.6)^0 = 1 * 1 = 1$ (el error cometido es $|0.625-1| = 0.375$)

iteración n = 1

$(-1)^0 (0.6)^0 + (-1)^1 (0.6)^1 = 1 - 0.6 = 0.4$ (el error cometido es $|0.625 - 0.4| = 0.225$)

iteración n = 2

$(-1)^0 (0.6)^0 + (-1)^1 (0.6)^1 + (-1)^2 (0.6)^2 = 0.4 + 0.36 = 0.76$ (el error cometido es $|0.625-0.76| = 0.135$)

iteración n = 3

$(-1)^0 (0.6)^0 + (-1)^1 (0.6)^1 + (-1)^2 (0.6)^2 + (-1)^3 (0.6)^3 = 0.76 - 0.216 = 0.544$ (el error cometido es $|0.625-0.544| = 0.081$)

Te pedimos:

Realizar un programa en ensamblador MIPS que calcule el número de términos a sumar para aproximarse a $\frac{1}{(1+x)}$ cometiendo un error igual o inferior a uno que se pida por consola.

Para ello deberás realizar las siguientes operaciones:

- 1) Solicitar el valor del número x por teclado, comprobando que el valor absoluto del mismo es inferior a 1. Si se introduce el número 0, el programa debe finalizar.

- 2) Solicitar el **error objetivo**, es decir, el máximo error que vamos a permitir para realizar la aproximación. Debe comprobarse que el error objetivo introducido por el usuario es mayor que cero y menor que el valor real de calcular $\frac{1}{(1+x)}$
- 3) Calcular el sumatorio en un bucle, añadiendo un término del sumatorio cada vez, y calculando el **error cometido**. El bucle debe terminar cuando el error cometido sea inferior al error objetivo.
- 4) Mostrar por consola, cual es el número de términos que se ha calculado (el valor de la n), y el error cometido en la aproximación.
- 5) Volver al paso (1), el programa finalizará cuando el usuario le dé un valor 0 a la x.

Antes que nada recuerda que debes realizar el pseudocódigo o código en lenguaje de alto nivel del ejercicio propuesto. Deberás ponerlo como comentarios en tu código en ensamblador junto con información sobre los registros que utilizas (esta manera de proceder te ayudará a ser ordenado en tu código y también a entender por parte del corrector tu programa en ensamblador). Ten en cuenta que son operaciones en punto flotante, por lo que tendrás que usar registros y operaciones del coprocesador.

Pondremos en el campus a tu disposición una página que realizará diferentes pruebas sobre tu código de forma automática. Para que funcione correctamente y supere el test es necesario que utilices las cadenas de textos que suministramos en el esqueleto, y que la ejecución de tu código funcione como te explicamos en el siguiente ejemplo de ejecución.

Ejemplo de ejecución:

```
PR3 aproximación por serie geométrica
Introduzca el valor de x ( $|x| < 1$ ): 2

Introduzca el valor de x ( $|x| < 1$ ): -1.4

Introduzca el valor de x ( $|x| < 1$ ): 0.6

Resultado real = 0.62500000
Introduzca el error objetivo: 0.001

Resultado calculado para 13 terminos = 0.62581623
Error cometido para 13 terminos = 0.00081623
Introduzca el valor de x ( $|x| < 1$ ): 0.6

Resultado real = 0.62500000
Introduzca el error objetivo: -1

Introduzca el error objetivo: 0.626

Introduzca el error objetivo: 0.0001

Resultado calculado para 18 terminos = 0.62493646
Error cometido para 18 terminos = 0.00006354
Introduzca el valor de x ( $|x| < 1$ ): 0

FIN DEL PROGRAMA.
|
```

Como puedes observar en el ejemplo de ejecución se comprueba que los valores de entrada son los correctos (cumplen con lo especificado en el enunciado), y si no lo fueran vuelve a repetir la pregunta para volverlos a introducir (se comprueba que el valor absoluto de la x es menor que uno, y que el error objetivo es positivo y menor que el resultado real). El programa finaliza cuando a x se le da un valor de cero.

Cuestiones:

- 1) Mejora tu programa, minimizando el número de operaciones que se realizan al calcular cada término.
- 2) Cuando se programa en ensamblador hay que elegir de manera adecuada los registros para realizar las operaciones. Di qué registros utilizaste para tus cálculos y cuál fue tu criterio para elegirlos.
- 3) Explica brevemente cómo puedes transformar tu programa a doble precisión. Escribe de forma detallada si realizar este cambio afecta a la elección de tus registros. Explica también si tiene algún efecto en el programa (sobre todo si mejora en algo los resultados).

Se entregará un programa fuente en ensamblador MIPS bien que contenga:

- En comentarios el pseudocódigo que utilizaste para resolverlo.
- En comentarios explica qué mejoras realizaste en el punto (1) para ahorrar cálculos.
- En comentarios deberá estar la respuesta al apartado (2).

- **En comentarios deberá estar la respuesta al apartado (3).**