

Estimating π — Application of the Leibniz Formula

Description:

The mathematical constant π can be approximated using the **Leibniz formula**, defined as:

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

write a C program that uses this formula to estimate the value of π and compare the results obtained with **float** and **double** data types.

Requirements:

1. Read an integer N from `in.txt`, representing the number of **terms** used for calculation.
2. Compute π using both:
 - o float precision
 - o double precision
3. Write the output into `out.txt`:
 - o π calculated with float
 - o π calculated with double
 - o Which result is more accurate

```
#define real_pi 3.1415926535
```

圓周率估計實驗 Leibniz 公式的應用

作業說明：

圓周率 π 是數學中最著名的常數之一，我們可以透過 Leibniz (萊布尼茲)

公式來計算它的近似值：

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

請撰寫一個 C 語言程式，利用此公式計算 π 的近似值，並比較 `float` 與

`double` 兩種資料型別的結果差異。

作業要求：

1. 程式需讀取 `in.txt`，代表使用前 N 項 (terms) 進行計算。
2. 使用 `float` 型別與 `double` 型別各自計算 π 。
3. 將答案讀進 `out.txt`：
 - 使用 `float` 計算出的 π
 - 使用 `double` 計算出的 π
 - 哪一種結果較為精確

```
#define real_pi 3.1415926535
```

Example I/O (1)

cat in.txt

100000

gcc hw7.c -o hw7

./hw7 <in.txt> out.txt

cat out.txt

[float] $\pi \approx 3.14158583$ | error = 0.00000683

[double] $\pi \approx 3.1415826536$ | error = 0.0000099999

=> float is more accurate (unusual!)

Please save error as abs number.

Example I/O (2)

cat in.txt

1000000

gcc hw7.c -o hw7

./hw7 <in.txt> out.txt

cat out.txt

[float] $\pi \approx 3.14159536$ | error = 0.00000271

[double] $\pi \approx 3.1415916536$ | error = 0.0000009999

=> double is more accurate (unusual!)

Please save error as abs number.