

## Programming language version

C++(使用 Dev C++寫成，Compiler : GCC 4.9.2)

## Implement Steps

使用 `fstream` 中的 `ifstream` 讀檔、`ofstream` 輸出檔，將 `U` 初始化為 `input` 的 `matrix`，將 `U` 當作 `A` 對自己做高斯消去法，進行 `LU` 分解，每做完一個 `row`，去檢查下一列的 `pivot` 值是否為 `0`(除了最後一個 `row` 不需要檢查)，如果 `pivot` 值為 `0`，則 `U` 不是 `upper triangular form`，判定為 `False` 並調整 `P、L` 的 `element`，最後依照 `True or False` 印出 `LU` 或 `PLU`(使用 `fixed` 以及 `iomanip` 中的 `setprecision()`來四捨五入至小數點後第二位)。

## How to execute your code

將 `input` 輸入至 `input.txt`

使用 Dev C++開啟 `lu_decomposition.cpp`

編譯並執行，即可在生成出的 `output.txt` 中看到 `output`

## Reference

網站: [cout 控制輸出的小數點位數](#)| [電腦不難](#)  
[LU 分解-維基百科](#)