**系統程式期中考（上機考部分）**

計分方式，請先從1.I及2.I選擇一題作為你的主答，主答為60分，之後每一題佔10分，最高100分。

1. Ron博士覺得蒙地卡羅方法使用了random函數，而random函數式系統的效能瓶頸所在，因此他想改用數值積分方法計算pi，數值積分方法是在X座標上分成n個等間距的點，接下來使用這n（介於0~1，包含0及1）個點產生n-1個矩形去逼近pi的值。矩形的畫法有二種，第一種採取左邊的線與圓的交叉點為高，畫出矩形（如圖中藍色的部分）。第二種採用矩形右邊的線與圓的交叉點為高，畫出矩形（如圖中紅色的部分）。由於藍色的部分的面積一定大於1/4圓，因此是上界（即圖中的upper bounds），而紅色的部分的面積一定小於1/4圓，因此是下界（lower bounds），透過上、下界我們就可以知道算出來的pi的準確度到小數點底下第幾位。
   1. （60pt/10pt）執行檔名稱為pi，執行方式為pi ###### $$，其中######為X軸上的等間距點的數量，$$為執行緒的數量。請輸出pi的上界及下界。（hint，上界和下界的矩形「剛好差一格」，你可以藉此來加速運算。換句話說下界的部分向右平移一個，再算上界的的最左邊一個，就可以得到上界）
   2. （10pt）令######一定是2的冪次方（例如：4096），以漸進的方式計算pi，例如：第一回合是8個點，第二回合是16個點。每當使用者按下ctr-c時印出當時已經計算出來的pi的值的上界及下界。使用者連續按下ctr-c（在2秒內按下二次）時終止程式並印出pi的值的上界及下界。請注意，使用漸進方式求pi時，你必須使用上一個回合的結果繼續往下推算。
   3. （10pt）執行pi程式時可以下達準確性參數，指令形式是 pi -pre 15，15表示pi必須準確到小數點底下第15個位數，請使用漸進的方式的算法計算pi，當使用者按下ctr-c時印出當時已經計算出來的pi的值的上界及下界。使用者連續按下ctr-c（在2秒內按下二次）時終止程式並印出pi的值的上界及下界。請注意，使用漸進方式求pi時，你必須使用上一個回合的結果繼續往下推算。
   4. 繳交：pi.c、makefile、readme.pi，在readme.pi中說明你完成了哪些功能，約略說明如何完成這些功能
   5. 協助你判斷程式的準確性，pi的近似值為：

3.141592653589793238462643383279502884197169399375105820974944592307816406286


Upper and Lower Bounds of area under a circle
	        

1. Ron博士想要擴充myshell.c的功能，請依照下列的要求修改myshell.c
   1. （60pt/10pt）當使用者按下ctr+\的時候，如果myshell正命令一個child在執行命令，那麼將ctr+\送給這個child，如果myshell並沒有叫child做任何事情，忽略ctr+\。（hint：ctr+\的signal代號是SIGQUIT）
   2. （10pt）Ron博士希望myshell.c支援FIFO的功能，請支援下面的命令
      1. mkfifo myfifo
      2. ls > myfifo & ，執行ls指令，並且將ls的stdout接到myfifo，由於這道指令最後面是&字元，因此請在背景執行ls > myfifo
      3. sort < myfifo ，執行sort指令，並且將sort的stdin接到myfifo

（hint：mkfifo是Linux的工具程式，因此你應該不需要特別的實作，上述命令都能在bash中使用，因此你可以先用bash觀察指令的行為）

* 1. 繳交：myshell.c、makefile、readme.fifo，在readme.fifo中說明你完成了哪些功能，約略說明如何完成這些功能