1. **題目說明:**Tamasulo algotithm實作
2. **使用函式庫:** <iostream> <string> <vector> <fstream>函數所在函式庫
3. **注意事項：**執行時間：add sub 2 cycle, mul 10 cycle, div 40 cycle

Issue&capture可在同一cycle執行

1. **函式介紹:**

1.vector<string> splitStr2Vec(string s, string splitSep)

根據指定的符號[:,()]將一行的字串做切割。

2. void tomasulo(string inst[][20],int address)

此次的作業我寫在同一個function，主要分成六大部分，分別是初始化、把值寫入RS、dispatch&放入buffer、capture、write result 、output。

1.初始化：

rf[]:int陣列，存rf裡面每個register的值

cycle:紀錄目前進行到第幾個cycle

current:紀錄目前讀取到第幾個inst

add\_rs[][]:紀錄add、sub的rs值

mul\_rs[10][6]: 紀錄mul、div的rs值

rat[6]:記錄每一個register目前對應的rat值

add\_bf[]:存add的buffer值

mul\_bf[]:存mul的buffer值

2. 把值寫入rs

讀取存在inst[][]陣列裡的第current行assembly code，並且判斷是做加減還是乘除，判斷完成後就會去做add or mul RS的選擇。

寫入的部分我是通過迴圈逐項確認每一個RS是不是空的，只要讀到是空的就去做寫入。

如果是要寫入F2那會先去看rat的第二項是不是空的，如果是空的就讀去rf第二項寫入後就透過break離開迴圈，便不會重複寫入了。寫入時，除了原本RS會存的項目，在add\_rs[]跟mul\_rs[]的第三項存要寫入到哪個register，如果要存到F1就存”1”，第五項我則存了寫入時是第幾個cycle，後面dispatch的時候會用到。

寫入後要去更新rat表，current也要+1。p.s. current 用來紀錄目前讀取到第幾個inst

3. dispatch&放入buffer

透過迴圈，逐項確認rs裡面的內容，如果rs裡面用來存值的兩項都不包含【R】(代表裡面的RAT已經代換成buffer算出來的數字)，且buffer目前是空的且目前的cycle不是issue進來的那個cycle，那就把值寫入buffer。Buffer裡面會存它是對應到RS幾、兩個要運算的數字、運算符號以及要存回哪一個register。

接著透過判斷運算符號後，會在buffer陣列的第五項存哪一個cycle可以做write result。陣列第六項則會存運算後的結果。

存入buffer後，原本存這些輸入的RS就可以清空了。

4. capture

如果buffer裡面現在存rs2，就透過迴圈逐項確認add\_rs跟mul\_rs裡面有沒有rs2，只要有的話就替換成buffer陣列里第六項的運算結果。

5.write result

如果buffer現在不是空的且現在的cycle是buffer陣列第五項存的可以做write result的那個cycle，那就去把buffer的值寫入RF，但是寫入前要確定一件事情。假設buffer 存rs2 是要寫到F2的，那要去RAT裡面確認，經過這麼多cycle之後，F2是不是還是存rs2，是的話才能去做寫入

寫入後也要記得把buffer清空！

6.output

就是不斷地cout，不多說明