1. **題目說明:** Input:一段RISC-V組語的codeOutput:對應的machine code
2. **使用函式庫:** <iostream> <string> <vector> <fstream>函數所在函式庫
3. **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.string inst\_to\_op(string inst[], int inst\_lth);

會根據字串的切割結果去判斷該行instruction的種類，並給予func3、func5、func2以及opcode等等預設的值，再呼叫函式去將register、immediate轉換成字串，最後合併所有傳換的結果return回main function。

3.string branch\_to\_op(string inst[], string Label[], int address,int label\_amount);

由於branch類的函式bne、beq等等會放Label名稱，故轉換的方式會跟普通instruction不同，故特別建立一個函式，會去計算address的位置差。

備註：我的jal只做最後面是immediate的形式，因為最後面是Label的做法是與beq,bne等等指令相同，故無重複再做。

4. string rg\_to\_string(string reg);

將register轉換成machine code的形式，再以string的型態保存，會去取register的數字部分，再透過運算轉換成二進位數。

5. string imme\_to\_string(string imme);

將immediate轉換成二進位形式存進string，包含正負數的作法，負數是利用計算出正數的二補數的方式做出(0010的2補數就是0010的flip再加上1)。

6. string ujimme\_to\_string(string imme);

Uj type要存的immediate形式比較不一樣(1-20位)故特別獨立一個function，作法與imme\_to\_string()類似。

1. **程式流程:**

我的作法是將組合語言寫在txt檔，再透過讀檔的方式將其讀入，透過vector<string> splitStr2Vec(string s, string splitSep)先去做字串切割，接著先跑第一次的while迴圈去確認Label的位置，並將address(預設從0開始，每行都+4)存進陣列，如果Label名稱叫L2，address是8，則我會儲存成L28，之後再去做切割。

第二次迴圈會去呼叫string inst\_to\_op(string inst[], int inst\_lth);去判斷指令的種類，並且將切割後的字串再透過其他函式轉換成opcode，結果回傳後將最後的答案合併再回傳回main function。