Input條件 1.register開頭必須為小寫x   
 2.label必須隔一行寫，最後必須是’:’字元  
 3.輸入完後必須隔一行輸入一行”-1”  
 4.輸入類別後要空白一格，之後所有register要連續輸入用’,’隔開  
 5.只可以輸入RV32I Base Integer Instruction Set裡的code(Fence除外)

Ex:

//Start  
loop:  
add x1,x2,x3  
beq x0,x0,loop  
sw x1,1000(x3)  
-1   
//End  
output會是每一行的machine code 每行32個bit

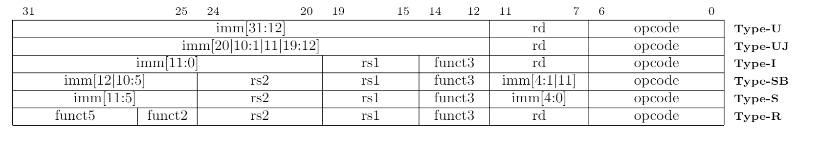
首先，讀取完所有code之後才做分析轉換成machine code

code 存放在 vector<string> code裡面，且label(最後是’:’)的有另外用 map<string,int>label來存有出現過這個label他在第幾行

另外每行code都會用空白來分割(label不算)所以透過stringstream ss來分割空白的兩邊，左邊放進string s右邊放進string b，之後分析s是哪個code去找對應的關鍵字，例如add就去add的那個if裡面繼續做下去，之後因有規定輸入格式，所以可以透過s是啥知道b的詳細格式，藉此分析出register有些啥，immediate跟offset又是多少，branch就透過分析完後得知是register有啥跟要跳到哪個label，而因為在分析這行code的時候是從vector<string>code裡面讀出來的，所以可以知道現在是第幾行code，再去透過map<string,int>label裡面找出這個label是在第幾行之後兩個去做相減找出branch類型的immediate要放啥

另外在找出相關register跟offset還有immediate之後，會透過offsetToBinary還有DecimalToBinary轉成一個二進位的表示，只是是string型態的，回傳回去存放到string a[ ]裡面，之後當作傳進各個type的function的parameter

之後根據



把所有參數放進要拿來輸出machine code的string BinaryOutPut裡面

最後再輸出machine code 也就是BinaryOutPut

剩餘詳細請見code的註解