解釋程式:

在移動方面，我的做法跟上次一樣。只是多了字串比較和超過範圍的判斷。

字串比較我在課本上查到一個很好用的procedure叫做Str\_Compare，這個procedure要搭配ADDR這個operator一起使用。可以用來比較字串，當字串相等時，carry flag會被設成0 zero flag則會被設成1，可以用JE來判斷是否要跳走。

超越範圍就輸出error訊息，然後跳到迴圈底部，準備直行下一個迴圈。然後比較困難的是判斷撿到的寶藏上面。我用的方法是，把這次移動的範圍記起來，超越這個範圍就跳走，沒超越得就去Check之前這寶藏有沒有被撿過，如果沒有的話，就把他記起來，之後輸出。以向右移動的為例: 把原本位置(temp reg)和移動過後位置(esi)記起來，去比較。首先Y軸要一樣，然後把寶藏一個一個拿出來比，比移動後位置大就跳走，比原本位制小也跳走。這樣跑下來的就是會找到的寶藏。然後再Check 方面，我是把先前找到的寶藏是第幾個寶藏記在treatureFlag陣列裡，這樣之後一個一個比較，如果沒有找到一樣的，就可以當作找到。

例如:

0 4 8…..

treatureX 1 2 3 ….

treatureY 0 1 3….

我現在假設從(0,0)走到(1,0)這個位置，就會把treatureX和treatureY一個一個拿近來判斷是否為找到，可以發現第0個位置就是，所以去呼叫check判斷，如果沒有，就把0記錄在treatureFlag裡。

之後就只要把路徑和總共拿到的寶藏印出來即可。此時只要小心指有移動0步或是error(超出範圍的)不該記錄。

心得和遇到的困難:

我這次花了蠻多時間的。且大約有九成五的時間花在DEBUG上，這次寫了很多的jump，常常跳來跳去把自己搞得頭昏眼花，最後只好按F11一個一個指令追蹤。這次程式在想法上或許不難，但組語有些東西不加註解隔一段時間還真的蠻難看懂的~~，所以這次我幾乎只要是容易看不懂的都有加註解。然後我發現，把所有東西都擠在main裡面真的蠻難DEBUG的~~，應該要多多把一些東西分到其他procedure去，再去call他(不過千萬要記得寫ret)。然後就是以後應該不要把長期要存的東西擺在暫存器裡，像這次我把現在位置擺在X: esi，Y: edi 。應該要把他用一個memory記起來或許比較好，因為我覺得這樣常常在做一些運算時，少了幾個register可用，會造成一些麻煩。希望下次我可以做得更好!!