Data Structure-Homework4

A1115513 劉沛辰

1. 心得:

這次作業給我的挑戰蠻大的，一開始閱讀功課要求時完全不知道從何下手，

後來想了一下決定先做好BCD加法，回來做BCD加法讓我複習了位元運算，把後4位進行AND之後做OR，取得4位元做加法，得到carryin跟carryout，之後把後面16位都做完。

後面再做linked list，把物件設為剛剛做的BCD，這裡遇到很大的input問題，分割輸入的字串放入linked list，然後分別帶入之前做好的BCD ADD。

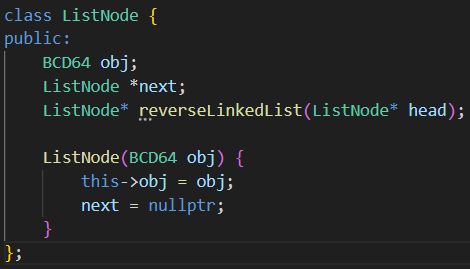
最後再進行reverse，讓他從高位輸出到低位。

1. The advantages of Big number with BCD encoding::

Linked list的好處就是好插入和增加元素，正好與大數運算一樣，因為無

法預測輸入的數字能有多大，所以用linked list做儲存，並且原本用array儲存就要做尋訪了，反而array的優勢(快速定位元素)的功能沒有顯現出來。

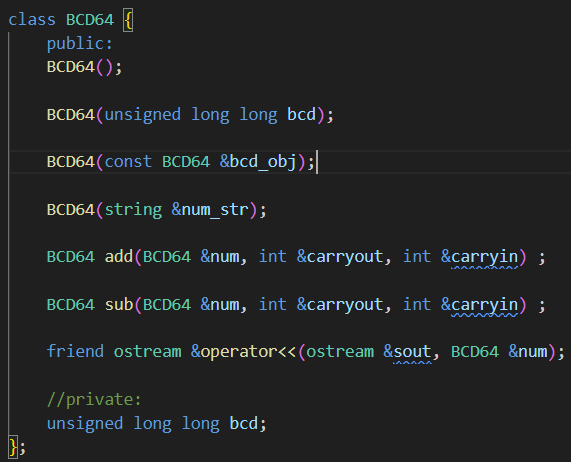
1. Design of your linked list:



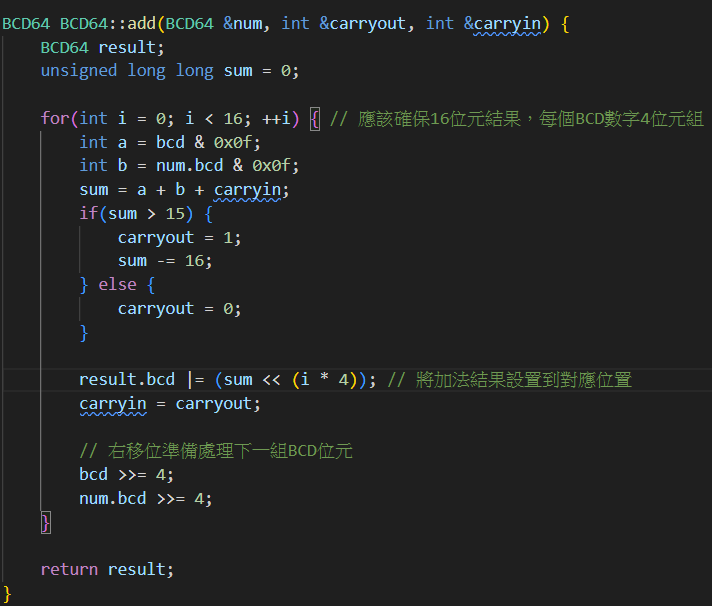
用自創的BCD64做為LinkedList的節點資料，還創建了一個reverse方便輸出(因為原本BCD處理是由低位到高位，輸出時要由高到低)。

1. Review/Improve class BCD64:

Review/Improve(此為更新過的):



定義多種constructor，讓使用者能輸入多種資料來建立，分別有空的，輸入int的，輸入BCD member的，和輸入string的，而下面定義了兩個function，分別是加法和減法，放入bcd、carryin，carryout，讓輸入的class bcd可以和原本的class bcd加減法



利用 and(位元運算子)，把原本unsigned long long分成16個part，分割的方法是用和 4b’1111做and的方式，取出最後4個bits，然後設一個sum做加法，當他overflow時，把sum-=16並把carryout設為1，然後把位元進行右移再繼續運算。

而減法則是sum用減的，當小於0把carryout設為1(借位)。

1. Comparison with other data structure:

使用python生成了不同長度的加法(資料由小到大排列)

使用unsigned long long int:











使用 int:







可以看到一開始差距不大，但是到後面因為分割的次數較多，造成延遲被放大，所以執行時間也變久