

加法：

用直式加法,假設有a b兩個BigNum物件,先宣告BigNum out,out的值就是a b同樣標號的值相加再加上out那格的值(預設0),如果得到的值後超過9的話out下一格的值就等於"此格的值/10",而此格的值就等於"此格的值%10",然後一直加下去

減法：

用直式減法,假設有a b兩個BigNum物件,先宣告BigNum out,在判斷a b的值何者較大(另外寫了壹個member function compare,這邊的大小是指不論正負的大小),將較大的複製一個物件為c,此時較小的在以下稱為d,out的值就是c d同樣標號的值相減(大的物件減小的)再加上out那格的值(預設0),如果在此格c的值小於d的值,那麼就在c的下一格減一,c此格加10,接著在做c-d的動作,一直減下去

乘法：

用直式乘法,假設有a b兩個BigNum物件,先宣告BigNum out,在判斷a b的值何者較大(另外寫了壹個member function compare,這邊的大小是指不論正負的大小),此時較大的在以下稱為c,較小的在以下稱為d,使用兩個for迴圈(以下外面的稱first,裡面的稱second),外面的(first)是由0至d的大小,裡面的(second)是由0至c的大小,out在first參數+second參數的值是原本的值(預設0)加上d[first]*c[second],如此一直乘下去,最後再做進位(如果out的值後超過9的話out下一格的值就等於"此格的值/10",而此格的值就等於"此格的值%10")

input >>：

直接讓他輸入到壹個string,然後宣告壹個BigNum out用這個string作為參數,然後讓進來的&in 等於 out,接著return o (ostream)

output <<：

先判斷如果長度為0的話,就直接輸出0然後return i (istream)

如果不為0的話,判斷如果sign=='-',就輸出 - .接著用壹個for迴圈將每一格的值照順序輸出,最後return i (istream)

prefix++:

就是先加一接著return自己的值,因為他要return的值是最後的值所以可以return reference

postfix++:

為了跟prefix區別,因此會多壹個參數

就是先return再++,但一旦return就無法做加一的動作,因此先宣告壹個BugNum out并將一開始的值複製過去,然後將原本的加一,最後return out

ps.因為out在function中宣告,所以離開後將被消滅,因此只能return value而不能return reference

負數:

我在加法、減法、以及乘法運作前都先用if做過判斷++ +- -+ --四種可能性,并將它們各別導到正確的位置,例如 『正數a 加上 負數b』,我會先宣告壹個tmp內容為b的值但宣告為正數,接著out=a-b,如此一來就可以將正負數都導入正確的位置,其中比較麻煩的是例如 『負數a 加上 負數b』,我會宣告tmp_1 tmp_2分別等於a b,且都宣告為正數,接著out=a+b,然後再將out的sign改為負數輸出