

物件導向程式設計

HW4

Report

0416037 李家安

1. how to implement +,-,* operator overloaded:

(a) +

在加法的部分，我先考慮被加數與加數是否為負，若其一為加數，則將其轉入減法實作，若同為正或同為負，先考慮是正或負，負的話將答案的 sign bit 指派為一，再各項分別相加，可是要考慮的是，若有進位，前一項加一，而本項模 10^5 (每項存五位數)。

(b) -

在減法的部分，考慮是否為一正一負，若是，則轉入加法，若否，檢查被減數與減數中較大，若減數較大則減數減被減數，並使 sign bit 變號，反之則直接減，若每一項有不夠減的，則前一項減一，此項加 10^5 。

(c) *

在乘法的部分，sign bit 以 xor 的方式處理，而數字部份，用兩層迴圈相互相乘，在將結果相加，放入 ans 裡，若有進位，記得作進位處理。

2. how to overload input and output operator

(a) output

先檢查 sign bit，若為一，輸出負號，之後的數字每五個五個印出，另外在 cout 時記得加入 right、setw(5)與 cout.fill("0")，纔可以準確印出每個數字。

(b) input

先以字串方式讀入，再以之前寫好的 constructor 將字串轉成大數。

3. What's difference between prefix and postfix increment overloaded?

Why they have different return type?

(a) difference

prefix 會直接將數字加(減)一，再將之輸出，而 postfix 則是另開一個大數，將數字存入，原來數字加(減)一，再回傳另開的大數。

(b) return type

prefix 的回傳值是以參照，因為他不用另開大數，因此可以直接回傳

原值以增加速度，另外也可以有在外部修改其值的功能，而此回傳值也不會在跳出函式後被 delet 掉，而 postfix 因為要另開大數，若以參照回傳，則回傳值將會在跳出函式後被 delet 掉，而產生 memory error，因此只能用一般(copy)方式回傳。

4. how to handle negative big number operation

對於負數，我開一個 bool sign;來記錄，當 sign 為 true 時，數字為負，反之則為正，之後只要檢查 sign 便可知其正負號了。