Data Structure Homework 5

繳交期限：　　2020/11/17　17:00 前
補交期限(7 折)：2020/11/24　17:00 前

手寫題：

2. Imagine we have the list shown in Figure 5-26 implemented as a linked list. As discussed in “List Search,” in Section 5.2, the search needs to be able to pass back both the location of the predecessor (`pPre`) and the location of the current (`pLoc`) node based on search criteria.

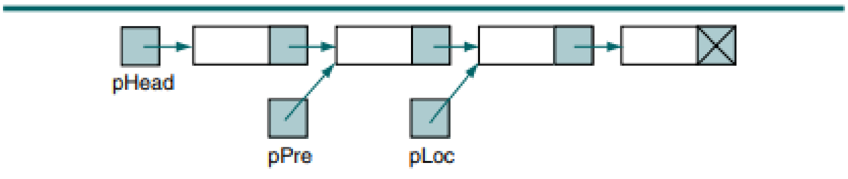


FIGURE 5-26 Linked List Implementation for Exercise 2

The following code to set `pPre` and `pLoc` contains a common error. What is it and how should it be corrected?

```
pLoc = pLoc->link
pPre = pPre->link
```

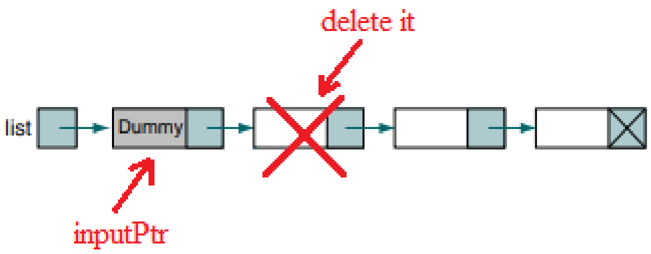
(*Hint:* What are the contents of these pointers at the beginning of the search?)

手寫 5-2 補充說明：(請詳敘)

1)請說明程式碼哪些部分有錯誤　，為什麼會有 error　？

2)並且將程式碼更正成正確的

4. Write the code to delete a node in the middle of a list implemented as a linked list with the dummy node (see Exercise 3). Compare your answer with the answer to Exercise 3. Are they the same? What do you conclude? Does the dummy node simplify the operation on a list? How?



請用 c 或 c++語法完成以下 function 使其完成上圖中的操作。

```
void deleteNode(Node* inputPtr) {
        ...
}
```

Link 是 Node 的 member，為指向 List 中的下一個 Node 的 pointer。

請用手寫並在 20 行內完成。

6. Write the statements to add a node in the middle of a list with the dummy node (see Exercise 3). Compare your answer with the answer to Exercise 5. Are they the same? What do you conclude? Does the dummy node simplify the operation on a list? How?

8. What would happen if we applied the following statements to the two lists in Exercise 7?

```
1 set temp to listl
2 loop (temp link not null)
  1 set temp to temp link
3 end loop
4 set temp link to list2
```

程式題：

26. Write a program to read a list of students from a file and create a list. The program should use a linked list for implementation. Each node in the linked list should have the student's name, a pointer to the next student, and a pointer to a linked list of scores. There may be up to four scores for each student. The structure is shown in Figure 5-33.

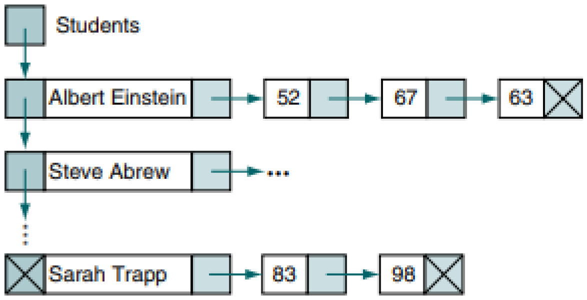


FIGURE 5-33 Data Structure for Project 26

The program should initialize the student list by reading the students' names from the text file and creating null score lists. It should then loop through the list, prompting the user to enter the scores for each student. The scores' prompt should include the name of the student.

After all scores have been entered, the program should print the scores for each student along with the score total and the average score. The average should include only those scores present.

The data for each student are shown in Table 5-3.

Student name	Score 1	Score 2	Score 3	Score 4
Albert Einstein	52	67	63	
Steve Abrew	90	86	90	93
David Nagasake	100	85	93	89
Mike Black	81	87	81	85
Andrew Dijkstra	90	82	95	87
Joanne Nguyen	84	80	95	91
Chris Walljasper	86	100	96	89
Fred Albert	70	68		
Dennis Dudley	74	79	77	81
Leo Rice	95			
Fred Flintstone	73	81	78	74
Frances Dupre	82	76	79	
Dave Light	89	76	91	83
Hua Tran	91	81	87	94
Sarah Trapp	83	98	94	93

TABLE 5-3 Data for Project 26

Input:

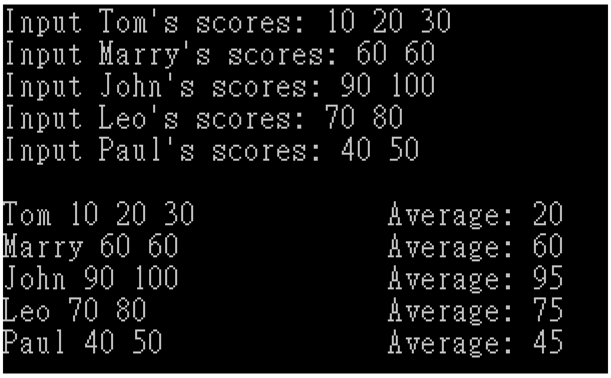
File：請參考附上的 5_26_input.txt，並在程式內改成讀取 input.txt，測資最多為 10 位學生。

Score：請讓使用者依學生的順序輸入成績(最多 4 筆)，不用依學生姓名來輸入，成績只會是 0~100 之間的整數(包含 0 和 100)，並用空白鍵隔開，輸入 Enter 後換成輸入下個學生的成績。

Output:

將學生姓名、各個分數印出，最後印出所有分數的平均

Input/Output example:



```
Input Tom's scores: 10 20 30
Input Marry's scores: 60 60
Input John's scores: 90 100
Input Leo's scores: 70 80
Input Paul's scores: 40 50

Tom 10 20 30            Average: 20
Marry 60 60             Average: 60
John 90 100             Average: 95
Leo 70 80               Average: 75
Paul 40 50              Average: 45
```

評分標準：

1. 是否使用 linked list (5 分)
2. 平均是否正確(須為 float) (3 分)
3. 資料數量是否正確、有無少資料 (1 分)
4. 程式無法執行則只給 1 分

注意事項：

1. 成績為 0 和沒有成績是不同 case，平均的計算也不同
2. 請將程式命名為"學號_姓名_hw5_26.cpp"or"學號_姓名_hw5_26.c"。
3. 不需要壓縮。
4. 只能繳交一個檔案，請將所有程式碼寫在一個檔案中。

29. Write a program that adds and subtracts polynomials. Each polynomial should be represented as a list with linked list implementation. The first node in the list represents the first term in the polynomial, the second node represents the second term, and so forth.

Each node contains three fields. The first field is the term's coefficient. The second field is the term's power, and the third field is a pointer to the next term. For example, consider the polynomials shown in Figure 5-34. The first term in the first polynomial has a coefficient of 5 and an exponent of 4, which is then interpreted as $5x^4$.

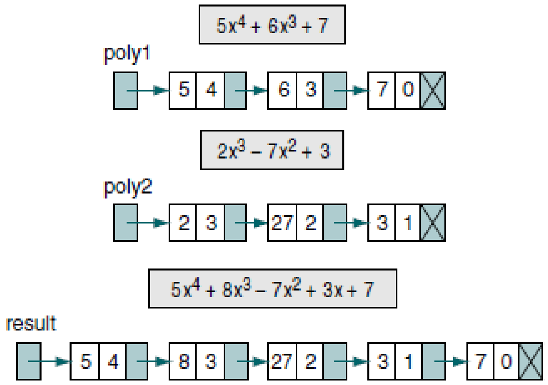


FIGURE 5-34 Example of Polynomials for Project 29

The rules for the addition of polynomials are as follows:

a. If the powers are equal, the coefficients are algebraically added.
b. If the powers are unequal, the term with the higher power is inserted in the new polynomial.
c. If the exponent is 0, it represents x^0, which is 1. The value of the term is therefore the value of the coefficient.
d. If the result of adding the coefficients results in 0, the term is dropped (0 times anything is 0).

A polynomial is represented by a series of lines, each of which has two integers. The first integer represents the coefficient; the second integer represents the exponent. Thus, the first polynomial in Figure 5-34 is

```
5    4
6    3
7    0
```

To add two polynomials, the program reads the coefficients and exponents for each polynomial and places them into a linked list. The input can be read from separate files or entered from the keyboard with appropriate user prompts. After the polynomials have been stored, they are added and the results are placed in a third linked list.

The polynomials are added using an operational merge process. An operational merge combines the two lists while performing one or more operations—in our case, addition. To add we take one term from each of the polynomials and compare the exponents. If the two exponents are equal, the coefficients are added to create a new coefficient. If the new coefficient is 0, the term is dropped; if it is not 0, it is appended to the linked list for the resulting polynomial. If one of the exponents is larger than the other, the corresponding term is immediately placed into the new linked list, and the term with the smaller exponent is held to be compared with the next term from the other list. If one list ends before the other, the rest of the longer list is simply appended to the list for the new polynomial.

Print the two input polynomials and their sum by traversing the linked lists and displaying them as sets of numbers. Be sure to label each polynomial.

Test your program with the two polynomials shown in Table 5-4.

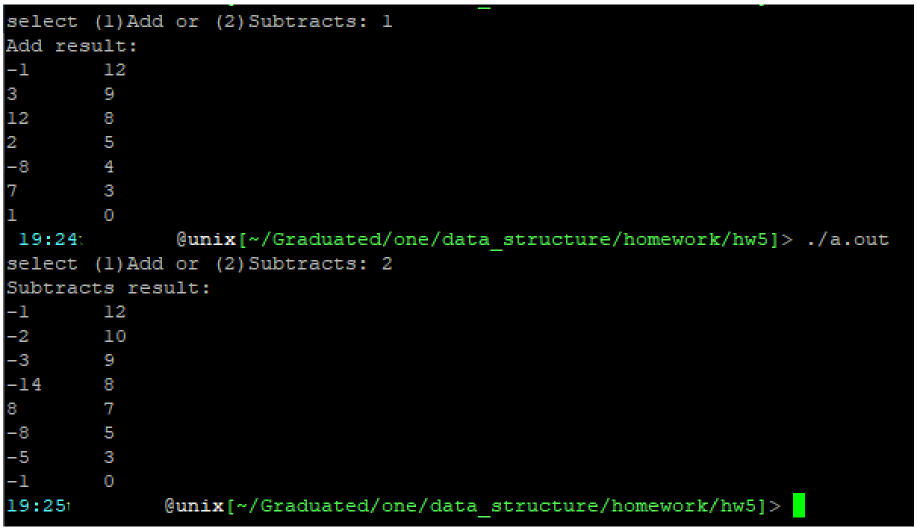
Polynomial 1		Polynomial 2	
Coefficient	Exponent	Coefficient	Exponent
7	9	–7	9
2	6	2	8
3	5	–5	7
4	4	2	4
2	3	2	3
6	2	9	2
6	0	–7	1

TABLE 5-4 Text Data for Project 29

程式 5-29 補充說明：

1)程式要有可以選擇多項式的加法和減法的功能

示意圖：



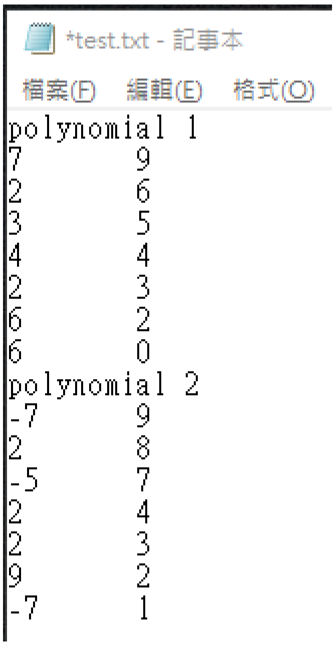
```
select (1)Add or (2)Subtracts: 1
Add result:
-1      12
3       9
12      8
2       5
-8      4
7       3
1       0
 19:24         @unix[~/Graduated/one/data_structure/homework/hw5]> ./a.out
select (1)Add or (2)Subtracts: 2
Subtracts result:
-1      12
-2      10
-3      9
-14     8
8       7
-8      5
-5      3
-1      0
19:25         @unix[~/Graduated/one/data_structure/homework/hw5]>
```

不用考慮重複輸入 (有或沒有都可以)

只會輸入 1 或 2 去選擇加法或減法，不用考慮其他輸入可能

2)測資請用 txt 檔讀取，檔名為"test.txt"

(勿由螢幕輸入多項式)



```
polynomial 1
7         9
2         6
3         5
4         4
2         3
6         2
6         0
polynomial 2
-7        9
2         8
-5        7
2         4
2         3
9         2
-7        1
```

3)使用 Table 5-4 中所示的兩個多項式測試您的程序(如上圖)

4)另外，助教會另外自行準備測資”test.txt” 下去測 (不是上面 result 圖)

(只會有兩個多項式運算，exponent 的範圍從 0 次~20 次都有可能)

5)結果顯示輸入在螢幕上 (如上 result 圖)

(勿輸出 txt 檔)

(6)其餘規則和題目相同

7)檔名為"學號_姓名_hw5_29.cpp"or"學號_姓名_hw5_29.c"，不用把測資 txt 檔上傳，不需要壓縮檔案。

只需繳交一個檔案，請將所有程式碼寫在一個檔案中

30. In older personal computers, the largest integer is 32,767 and the largest long integer is 2,147,483,647. Some applications, such as cryptography and security algorithms, may require an unbounded integer. One way to store and manipulate integers of unlimited size is by using a linked list. Each digit is stored in a node of the list. For example, Figure 5-35 shows how we could store a five-digit number in a list.

Although the list in Figure 5-35 is represented as moving from right to left, there is no physical direction in a list. We represent it in this way to clarify the problem.

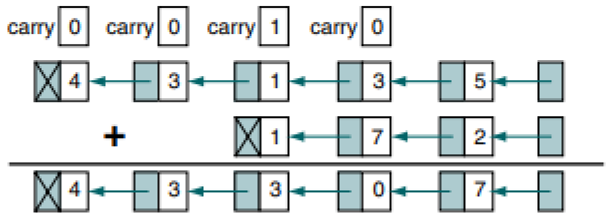


FIGURE 5-35 Integer Stored in a List for Project 30

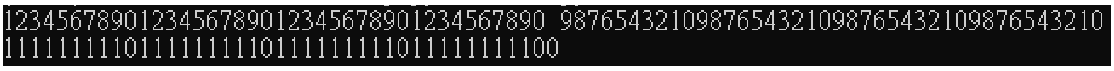
To add two numbers, we simply add the corresponding digit in the same location in their respective lists with the carry from the previous addition. With each addition, if the sum is greater than 10, we need to subtract 10 and set the carry to 1. Otherwise, the carry is set to 0.

Write an algorithm to add two integer lists. Design your solution so that the same logic adds the first numbers (units position) as well as the rest of the number. In other words, do not have special one-time logic for adding the units position.

說明：

輸入兩個不限長度的數字，輸出兩個數的總和。

範例：

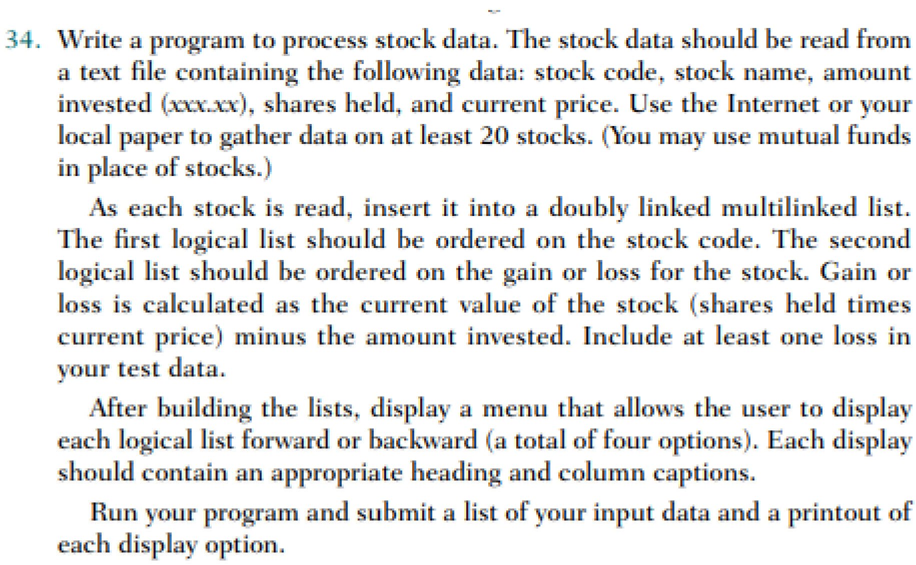


```
1234567890123456789012345678901234567890 9876543210987654321098765432109876543210
11111111101111111110111111111011111111100
```

問題與注意事項：

1. 請確保你的程式能被 g++或 gcc 編譯(使用 visual c++的請特別注意不要使用 g++或 gcc 中沒有的函式庫)
2. 數字必須使用 Link List 儲存，但不能使用 List 的函式庫，請自行撰寫。
3. 輸入格式：
   (1) 輸入用一個空白隔開。
   (2) 輸入數字不考慮負數。
   (3) 不考慮輸入錯誤(非數字字元)。
   (4) 輸入數字沒有長度上限，只要記憶體還沒爆就必須能正常執行。
   (5) 不需要任何提示文字，請不要輸出任何多餘的字元。
   (6) 程式只需要執行一次，不需要使用迴圈多次輸入。
   (7) 不符合以上格式者直接扣 2 分。

4. 繳交格式：
   (1) 請將程式命名為"學號_姓名_hw5_30.cpp"or"學號_姓名_hw5_30.c"。
   (2) 不需要壓縮。
   (3) 只能繳交一個檔案，請將所有程式碼寫在一個檔案中。
   (4) 不符合以上格式者直接扣 2 分。
5. 以上規範只限於此題中，其他請參考各題的說明。



34. Write a program to process stock data. The stock data should be read from a text file containing the following data: stock code, stock name, amount invested (*xxx.xx*), shares held, and current price. Use the Internet or your local paper to gather data on at least 20 stocks. (You may use mutual funds in place of stocks.)

As each stock is read, insert it into a doubly linked multilinked list. The first logical list should be ordered on the stock code. The second logical list should be ordered on the gain or loss for the stock. Gain or loss is calculated as the current value of the stock (shares held times current price) minus the amount invested. Include at least one loss in your test data.

After building the lists, display a menu that allows the user to display each logical list forward or backward (a total of four options). Each display should contain an appropriate heading and column captions.

Run your program and submit a list of your input data and a printout of each display option.

說明事項：

可以上網查詢現在台股的資料，會有股票代碼、公司名稱以及現在價格，自己選至少 20 家公司，可以自訂投資金額以及買入的股票張數，統一規定打成 txt 檔，再進行讀取，但是注意至少要有一家公司股票賣出後是賠錢的。

特別注意!! 要把程式檔和 input data 的 txt 檔壓縮成 zip 檔後上傳，格式為:
姓名_學號_5.34.zip(ex:王小明_309513068_5.34.zip)