

Paperwork

編譯並執行

此作業已經寫好 makefile，請使用 `make` 編譯。

Install make (Ubuntu 20.04)

```
1 $ sudo apt-get install make
2 $ sudo apt-get install build-essential
```

main.c

路徑 `./HW3/` 下 `make`。

```
1 $ make clean
2 $ make
3 $ ./main
```

使用者介面介紹

一個互動式 terminal 介面，過程中文字有上顏色以容易區分。

黃色：選項使用的引導以及提示

紅色：Error message

綠色：結束程式

示意圖：

```
HW3 : zsh — Konsole
檔案(F) 編輯(E) 檢視(V) 書籤(B) 設定(S) 說明(H)
~/De/com/N/HW3 make clean ✓ 03:40:20 下午
rm -rf main main.o name_list.o poly_parser.o
~/De/com/N/HW3 make ✓ 03:40:21 下午
gcc -Wall -Wextra -std=c11 -O2 -c -o main.o main.c
gcc -Wall -Wextra -std=c11 -O2 -c -o name_list.o name_list.c
gcc -Wall -Wextra -std=c11 -O2 -c -o poly_parser.o poly_parser.c
gcc -Wall -Wextra -std=c11 -O2 -o main main.o name_list.o poly_parser.o -lm
~/De/com/N/HW3 ./main ✓ 03:40:22 下午
Please enter a number 0~9 to choose what you want to do.
(0) Exit.
(1) Enter a polynomial of X (use the form of ax^b+cx^d...).
(2) Print all polynomials on the screen.
(3) Enter a degree, and show the coefficient.
(4) New a non-zero item in the polynomail.
(5) Remove a non-zero item in the polynomail.
(6) Show the result of a polynomial plus a polynomial
(7) Show the result of a polynomial minus a polynomial
(8) Show the result of a polynomial multiply a polynomial
Number: 1
Enter a polynomial of X (use the form of ax^b+cx^d+...+n).
5x^2+10+5x^-10-8x^100+10x^99
This polynomial name is P1
Please enter a number 0~9 to choose what you want to do.
(0) Exit.
(1) Enter a polynomial of X (use the form of ax^b+cx^d...).
(2) Print all polynomials on the screen.
(3) Enter a degree, and show the coefficient.
(4) New a non-zero item in the polynomail.
(5) Remove a non-zero item in the polynomail.
(6) Show the result of a polynomial plus a polynomial
(7) Show the result of a polynomial minus a polynomial
(8) Show the result of a polynomial multiply a polynomial
Number: 999
Error option!
Please enter a number 0~9 to choose what you want to do.
(0) Exit.
(1) Enter a polynomial of X (use the form of ax^b+cx^d...).
(2) Print all polynomials on the screen.
(3) Enter a degree, and show the coefficient.
(4) New a non-zero item in the polynomail.
(5) Remove a non-zero item in the polynomail.
(6) Show the result of a polynomial plus a polynomial
(7) Show the result of a polynomial minus a polynomial
(8) Show the result of a polynomial multiply a polynomial
Number: 0
Good Bye.
~/De/com/N/HW3 ✓ 8s 03:40:32 下午
```

使用方法

輸入 0 ~ 8 以選擇操作功能，如果輸入非 0 ~ 8 的數字會輸出 Error Message 並要求重新輸入。

功能介紹

1. 新增多項式以及命名

Procedure

輸入一字串並使用 `poly_parser` 分成 `coefficient` 與 `degree` 儲存至 `array`，並且使用 `qsort` 依照 `degree` 大至小排序，使用其儲存的 **P位置+1** 作為該多項式名稱。（如 P3）

使用規定

- 單項輸入規則：
 - 高次方：數字 $x^{\text{數字}}$ 。（Ex. $12x^{-345}$ or $-34x^{567}$ ）
 - 一次方：數字 x 。（Ex. $12x$ or $-34x$ ）
 - 常數：數字。（Ex. 123 ）
- 數字屬於整數（正負號都可）
- 單項間可使用 $+$ 或 $-$ 連接
- 以上輸入皆無空格

Error Message

無

Time Complexity

1. `poly_parser` 將字串掃過去並且將數字分開和儲存，因此整體複雜度為 $O(\text{該字串長度})$ 。
2. `query` 多項式的名字時只是將其 `array` 的 位置 + 1 並輸出，複雜度為 $O(1)$ 。
3. `qsort complexity`: The C standard doesn't talk about its complexity of `qsort`. 但是 worst case 會退化成 $O(n^2)$ ，否則是 $O(n \log_2 n)$ 。

2. 印出所有多項式

Procedure

將全部已輸入的多項式印出，遇到係數項為 0 則略過。（註記：多項式儲存方式為陣列裡儲存指數與係數，有係數項為 0 是下面基本操作刪除指定項的操作）

使用規定

無

Error Message

無

Time Complexity

$O(\text{多項式數量} \times \text{多項式長度})$ 。

3. 指定多項式 P 之指數次方，顯示該項的係數值

Procedure

從該字串的所有儲存 degree 值中找到 query 值。

使用規定

輸入 P 數字（中間沒有空格分開）。（Ex. P3）

Error Message

找不到「多項式 P 或 query 值」會報錯。

Time Complexity

將多項式 P 從頭至尾進行詢問，為**線性複雜度**。

4. 可新增 / 移除多項式

Procedure

- 新增
 - 如果儲存空間中沒有係數項為 0 的，則將 array 多加一個空間並將係數與指數項存入，如果次方項重複，則覆寫該係數。前者由 qsort 排序，後者不會排序。
 - 反之就覆寫指數項為 0 的區域，並由 qsort 排序。
- 移除
 - 找到指定次方項將其係數歸 0。

使用規定

無

Error Message

- 新增：如果有重覆只是輸出 warning message 並將該項係數覆寫。
- 刪除：如果找不到該項會報錯。
- 找不到「多項式 P」會報錯。

Time Complexity

先搜尋指定項於該儲存係數與指數項的 array，為**線性複雜度**；再新增或歸 0。 $O(1)$ 。qsort 排序的部份第 1 點有詳細解釋。

5. 相加兩個多項式

Procedure

先將一多項式係數與指數項儲存的 array 複製到另一個 array，再以另一個多項式係數與指數項的 array 逐個尋找新 array 中重複的 degree 並將其係數相加，其餘沒有重複的增加至新 array 的後面，最後使用 qsort 排序後輸出。

使用規定

無

Error Message

- 找不到「多項式 P」會報錯。
- 多項式總數量 ≤ 1 會報錯。

Time Complexity

在複製到新 array，逐個尋找的過程，為 $P1_length + P1_length \times P2_length$
 $O(P1_length \times P2_length)$ 。
最後 qsort 排序的部份第 1 點有詳細解釋。

6. 相減兩個多項式

Procedure

先將一多項式係數與指數項儲存的 array 複製到另一個 array，再以另一個多項式係數與指數項的 array 逐個尋找新 array 中重複的 degree 並將其係數相減，其餘沒有重複的增加至新 array 的後面，最後使用 qsort 排序後輸出。

使用規定

無

Error Message

- 找不到「多項式 P」會報錯。
- 多項式總數量 ≤ 1 會報錯。

Time Complexity

在複製到新 array，逐個尋找的過程，為 $P1_length + P1_length \times P2_length$
 $O(P1_length \times P2_length)$ 。
最後 qsort 排序的部份第 1 點有詳細解釋。

7. 相乘兩個多項式

Procedure

將兩個多項式逐項相乘儲存至新的 array ，並且尋找相乘後的 degree 在新 array 中是否出現過，有的話相加，否則額外儲存。最後使用 qsort 排序後輸出。

使用規定

無

Error Message

- 找不到「多項式 P 」 會報錯。
- 多項式總數量 ≤ 1 會報錯。

Time Complexity

假設 n 為多項式的長度

多項式相乘並尋找相乘後的 degree 在新 array 中是否出現過，最差情況為 $P1_length \times P2_length \times (P1_length + P2_length)$ ， $O(n^2 \times (2n)) \rightarrow O(n^3)$ 。

最後 qsort 排序的部份第 1 點有詳細解釋。