|  |
| --- |
| 得 分 |
|  |

資料結構

HWK

兩方程式相加

B10702057盧昱達

* 程式碼

#include<iostream>

#include<fstream>

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

using namespace std;

typedef struct{

double c, x;

}Polynomial;

int toPolynomial(char a[100], Polynomial in[20]){ //strtok and atoi and retun n

int n = 0;

char \*sp, \*deter = " ";

in[0].c = atof(strtok(a, deter));

in[0].x = atof(strtok(NULL, deter));

n++;

while((sp = strtok(NULL, deter)) != NULL){

in[n].c = atof(sp);

sp = strtok(NULL, deter);

in[n].x = atof(sp);

n++;

}

return n;

}

void PrintPolynomial(Polynomial out[20], int n){

for(int i = 0; i < n - 1; i++){

if(out[i + 1].c > 0){

cout << out[i].c << "x^" << out[i].x << "+";

}else{

cout << out[i].c << "x^" << out[i].x ;

}

}cout << out[n - 1].c << "x^" << out[n - 1].x;

}

void InOpenDate(char CharIn[100], char AB){ //

fstream InF;

int n = 0;

char FName[20], ch; //FName = 檔名

cout << "輸入" << AB << "方程式檔名:\n";

cin >> FName;

InF.open(FName, ios::in);

if(!InF){

cout << "檔案無法開啟\n";

}else{

while(InF.get(ch)){

CharIn[n] = ch;

n++;

}

InF.close();

}

}

int addPolynomial(Polynomial a[20], Polynomial b[20], Polynomial c[20], int na, int nb){ //no GO

int nc = 0, an = 0, bn = 0;

while(an < na && bn < nb){

if(a[an].x > b[bn].x){

c[nc].c = a[an].c;

c[nc].x = a[an].x;

an++;

}else if(a[an].x < b[bn].x){

c[nc].c = b[bn].c;

c[nc].x = b[bn].x;

bn++;

}else if(a[an].x == b[bn].x){

c[nc].c = a[an].c + b[bn].c;

c[nc].x = a[an].x;

an++;

bn++;

}

nc++;

}

return nc;

}

int main( void ){

int na, nb, outn;

Polynomial inA[20], inB[20], out[20];

char ca[100], cb[100];

InOpenDate(ca, 'A');

InOpenDate(cb, 'B');

na = toPolynomial(ca, inA);

nb = toPolynomial(cb, inB);

PrintPolynomial(inA, na);

cout << " + ";

PrintPolynomial(inB, nb);

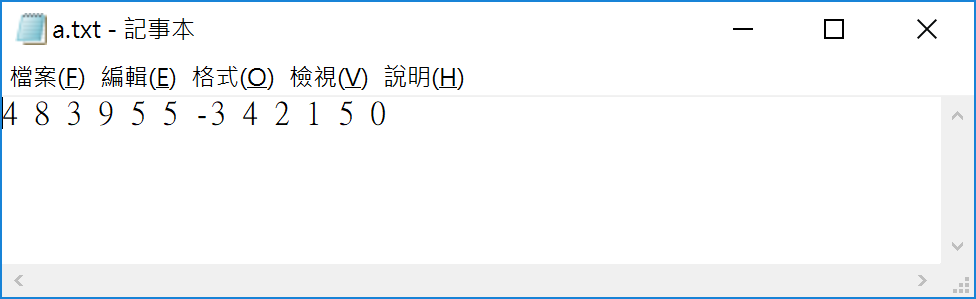
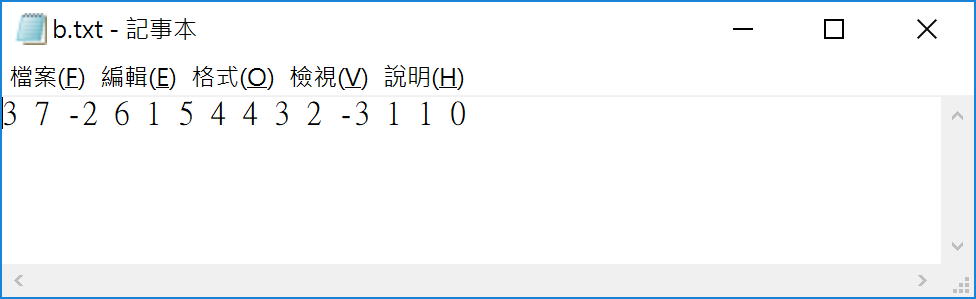
cout << " = ";

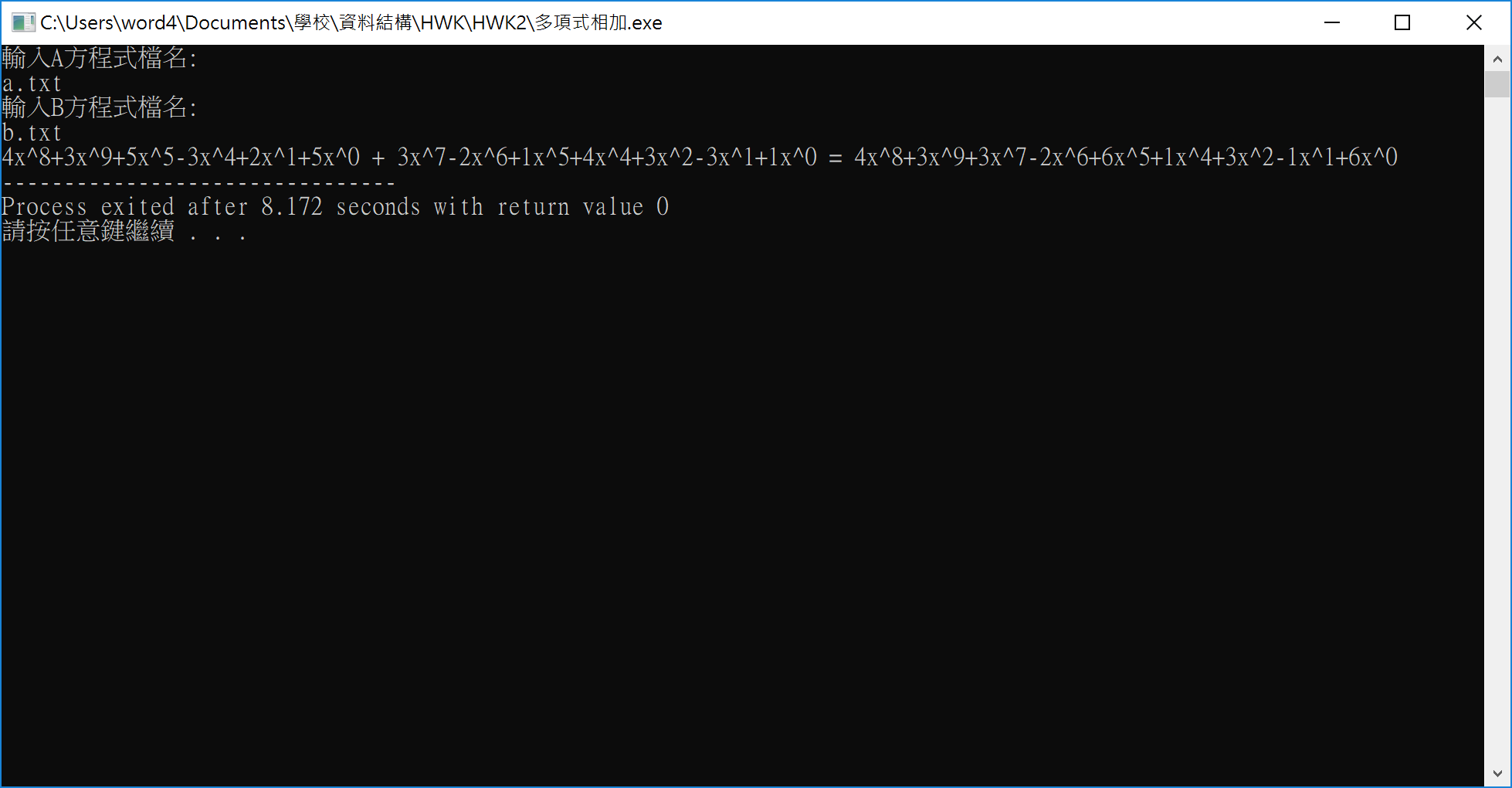
outn = addPolynomial(inA, inB, out, na, nb);

PrintPolynomial(out, outn);

}

* 測試結果



* 實作心得

我在實作這項功課時，發現如果用結構寫，將可以快速與清晰的紀錄參數，並在超作兩多項式相加時，更可以快速地達到目的。另外在這項功課中，也讓我複習到開檔與讀檔，是一道不錯的題目。