

嵌入式程式設計

1071 電機系

期中考報告

點到直線最小垂直距離和
與 Linear Regression 方法之程式

電機三 B

王靖嫻

B1055157

蕭培墉老師

目錄

一、 題目解析

二、 解題原理與數學

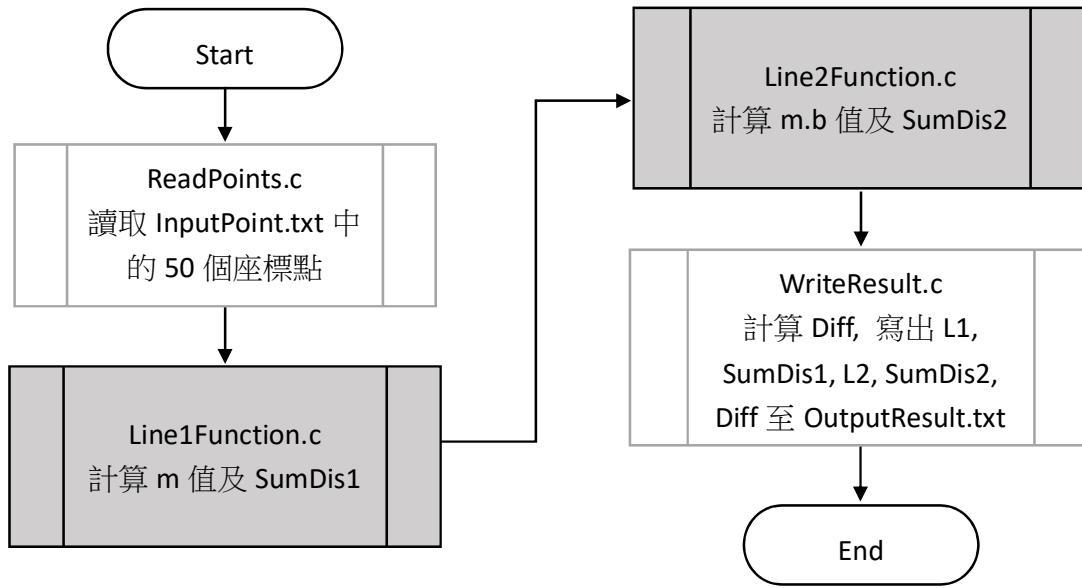
三、 演算法與流程圖

四、 其他設計文件

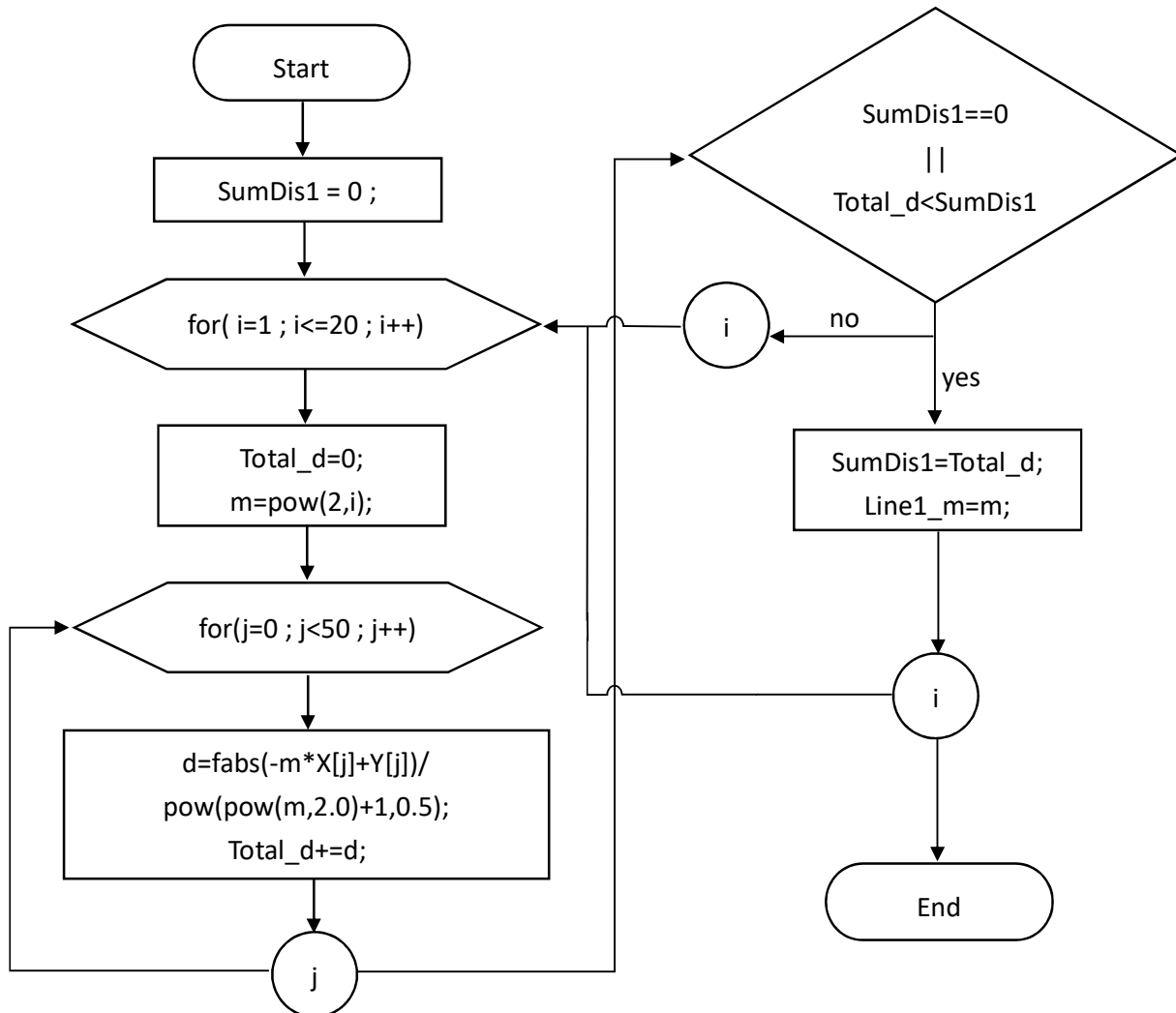
五、 心得與討論

六、 原始程式列印

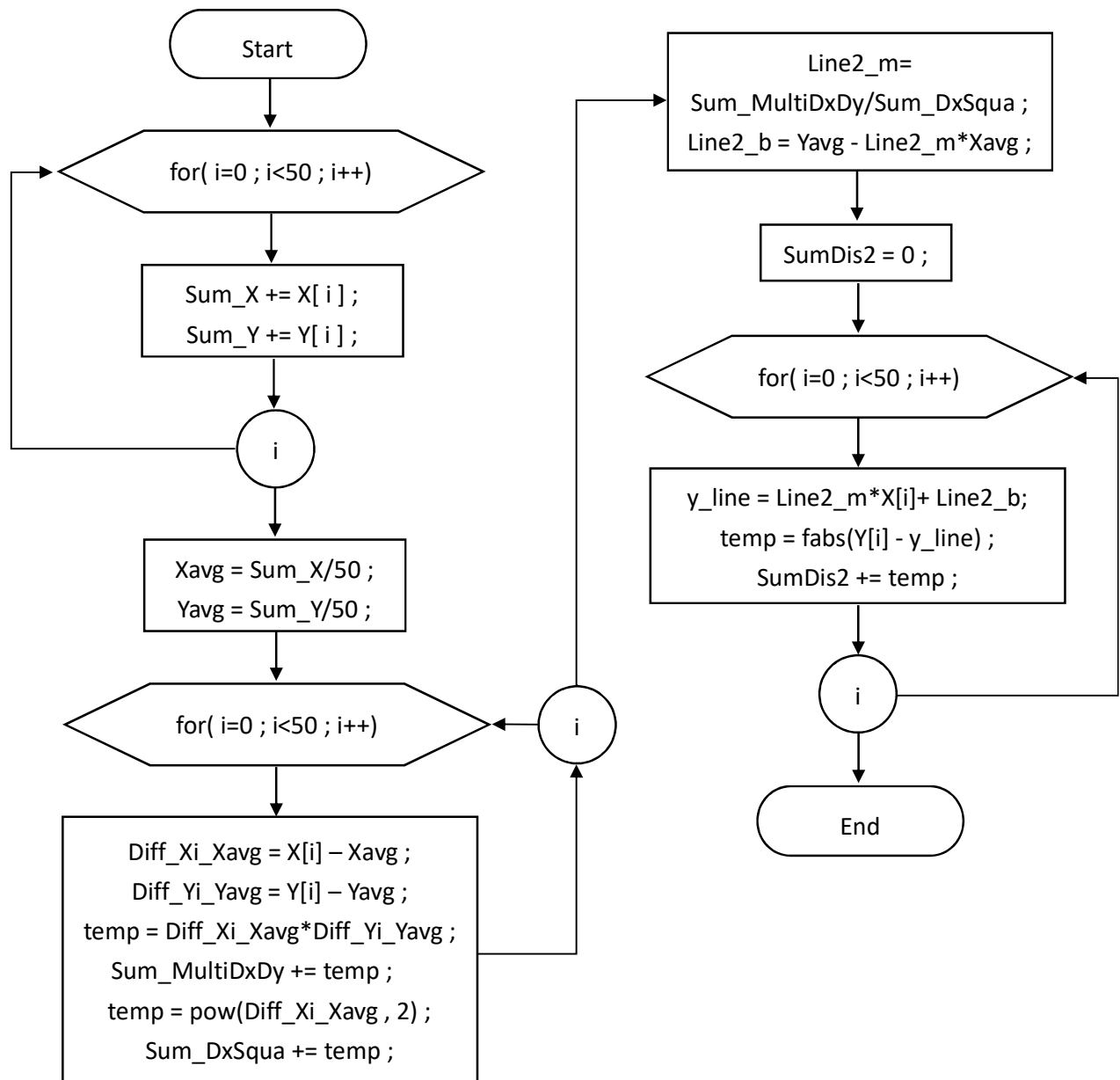
2-1. 主函式



2-2. 子函式 Line1Function.c

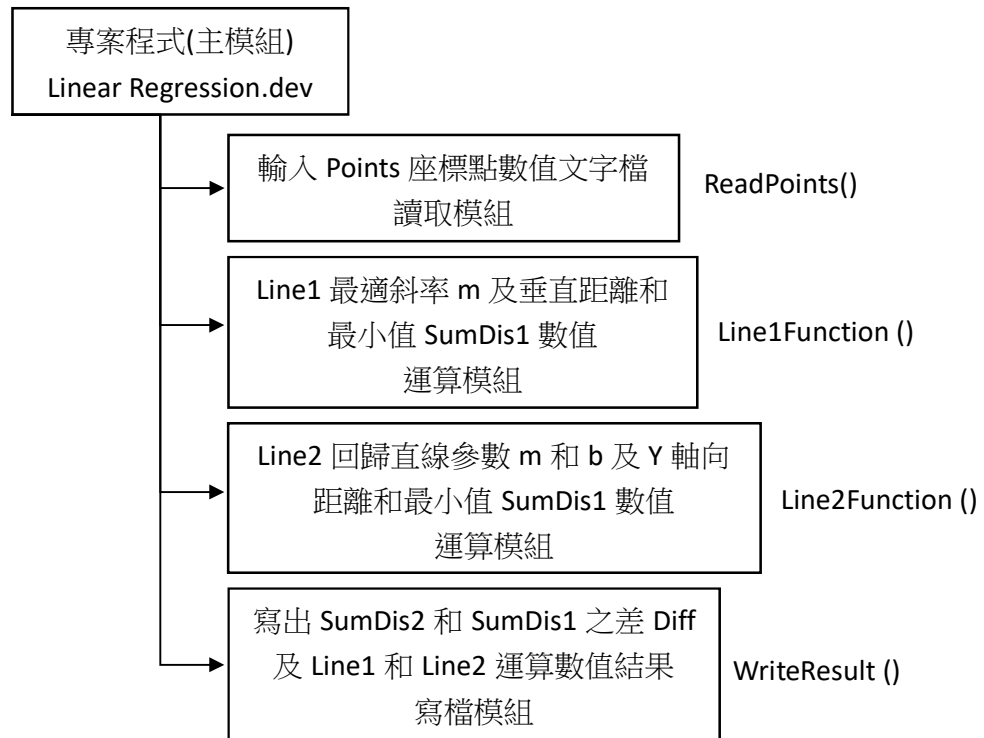


2-3. 子函数 Line2Function. c

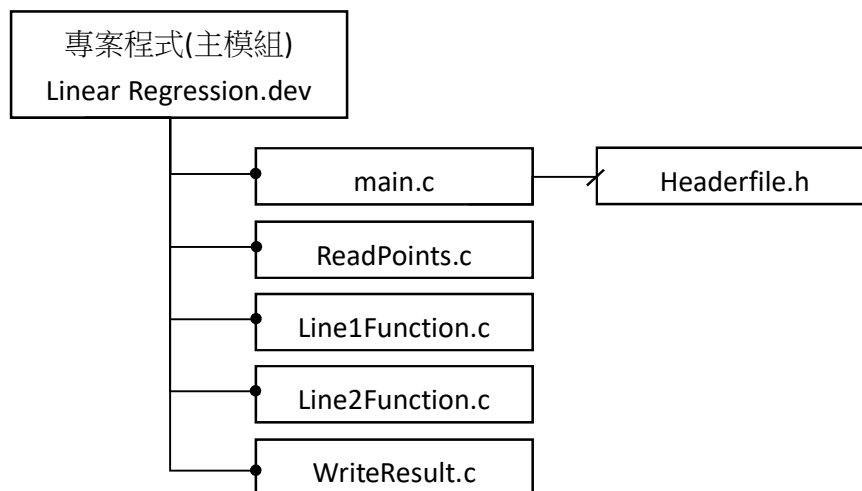


四、其他設計文件

1. 專案程式模組架構圖



2. 專案檔案關係圖



五、心得與討論

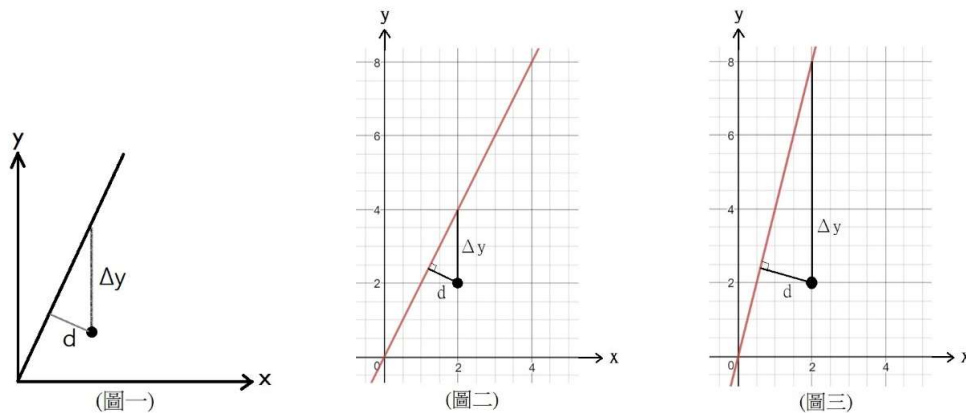
5-1. SumDis2 與 SumDis1 兩者大小差異所代表的物理意義之討論

● L2 y 截距值(b)大小觀點

由於 L1 被設定為通過原點之直線，而最適回歸直線(L2)已是能使 y 軸向距離和有最小值的直線，因此若某 50 個點所形成的 L2 之 b 值與 0 差距越大，L1 與最適回歸直線差距會越大，理應 SumDis1 越可能大於 SumDis2，但若 L2 之 b 值與原點差距越小，SumDis2 是有機會大於 SumDis1 的，解釋如下：

考慮極端 Case：若 50 個點所形成的 L2 最適回歸直線斜率恰為 L1 的斜率，且 y 截距為 0，則 L1 與 L2 重疊，由於 SumDis1 的取值為點到直線的垂直距離(d)之和，而 SumDis2 的取值為點到直線的 y 軸向距離(Δy)之和，將一點與直線和 d、 Δy 畫在座標平面上如下，從(圖一)中可觀察到形成的直角三角形中， Δy 為斜邊，而 d 為一股，因此

$\Delta y > d$ ，其餘點皆然，於是 $\text{SumDis2} > \text{SumDis1}$ 。



● 直線斜率值(m)大小觀點

考慮極端 Case：同上述

Case1：若形成的 m 值較小($2 \leq m \leq 2^{20}$ 且較接近 2 時)，對於同一點而言， $\Delta y > d$ 且 Δy 與 d 差距較小，所以 $\text{SumDis2} > \text{SumDis1}$ 且 SumDis2 與 SumDis1 差距較小，如(圖二)所示。

Case2：若形成的 m 值較大($2 \leq m \leq 2^{20}$ 且較接近 2^{20} 時)，對於同一點而言， $\Delta y > d$ 且 Δy 與 d 差距較大，所以 $\text{SumDis2} > \text{SumDis1}$ 且 SumDis2 與 SumDis1 差距較大，如(圖三)所示。

- 綜合比較

綜合上述可得結論：

- (i) 當某 50 個點所形成的 L2 最適回歸直線之 b 值與 0 差距越小且 m 值越大，則 $\text{SumDis2} > \text{SumDis1}$ 可能性越大，於是 Diff 越可能大於 0。
- (ii) 當某 50 個點所形成的 L2 最適回歸直線之 b 值與 0 差距越大且 m 值越小，則 $\text{SumDis2} < \text{SumDis1}$ 可能性越大，於是 Diff 越可能小於 0。

5-2. 本次實作報告心得

藉由本次實作的過程中學到了如何讀入文字檔數值和寫出文字檔及使用模組化的程式設計方法，讓程式結構化與可讀性提高，也了解了如何使用計算數字的次方之函數 `pow()`，還有計算絕對值的函數之使用，區別出絕對值的計算分為計算整數絕對值函數 `abs()` 以及計算小數絕對值函數 `fabs()`，以及由此報告中更加思考了兩種距離值所代表的物理意義，而解釋物理意義的過程中實際上也是在磨練自己的論述能力，有了好的論述能力才能使作出的結論更加具有信服力。

六、原始程式列印

■ Headerfile.h

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  typedef struct{
5      int* point_x;
6      int* point_y;
7  }Point;
8  typedef struct{
9      float Line1_m;
10     float SumDis1;
11 }Line1;
13 typedef struct{
14     float Line2_m;
15     float Line2_b;
16     float SumDis2;
17 }Line2;
18 typedef struct{
19     Point *Point_ptr;
20     Line1 *Line1_ptr;
21     Line2 *Line2_ptr;
22 }Datastruct;
23 void ReadPoints(char IPTxtfname[],char OPTxtfname[],Point *Point_ptr);
24 void Line1Function(Point *Point_ptr,Line1 *Line1_ptr);
25 void Line2Function(Point *Point_ptr,Line2 *Line2_ptr);
26 void WriteResult(char OPTxtfname[],Line1 *Line1_ptr,Line2 *Line2_ptr);
```

■ main.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "Headerfile.h"
4  int main() {
5      char IPTxt_Filename[350]="InputPoint.txt";
6      char OPTxt_Filename[350];
7      Datastruct *Datastruct_DS=(Datastruct*)malloc(sizeof(Datastruct));
8      Datastruct_DS->Point_ptr=(Point*)malloc(sizeof(Point));
9      Datastruct_DS->Line1_ptr=(Line1*)malloc(sizeof(Line1));
10     Datastruct_DS->Line2_ptr=(Line2*)malloc(sizeof(Line2));
11
12     ReadPoints(IPTxt_Filename,OPTxt_Filename,Datastruct_DS->Point_ptr);
13     printf("--輸入座標點文字檔: %s\n",IPTxt_Filename);
14     printf("--輸出運算結果文字檔: %s\n",OPTxt_Filename);
15     printf("--輸入50個座標點:\n");
16     int i;
17     for(i=0;i<50;i++){
18         printf("(%3d ,%4d ) ",Datastruct_DS->Point_ptr->point_x[i],
19             Datastruct_DS->Point_ptr->point_y[i]);
20         if(i%5==4&&i!=0)
21             printf("\n");
22     }
23
24     Line1Function(Datastruct_DS->Point_ptr,Datastruct_DS->Line1_ptr);
25     printf("Line1: y=%.1fx\n",Datastruct_DS->Line1_ptr->Line1_m);
26     printf("SumDis1 = %.2f\n",Datastruct_DS->Line1_ptr->SumDis1);
27     Line2Function(Datastruct_DS->Point_ptr,Datastruct_DS->Line2_ptr);
28     printf("Line2: y=%.1fx+%.1f\n",Datastruct_DS->Line2_ptr->Line2_m,
29         Datastruct_DS->Line2_ptr->Line2_b);
30     printf("SumDis2 = %.2f\n",Datastruct_DS->Line2_ptr->SumDis2);
31
32     WriteResult(OPTxt_Filename,Datastruct_DS->Line1_ptr,
33         Datastruct_DS->Line2_ptr);
```



```

34     free(Datastruct_DS->Point_ptr);
35     free(Datastruct_DS->Line1_ptr);
36     free(Datastruct_DS->Line2_ptr);
37     free(Datastruct_DS);
38     return 0;
39 }

```

■ ReadPoints.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "Headerfile.h"
4  void ReadPoints(char IPTxtfname[],char OPTxtfname[],Point *Point_ptr)
5  {
6      FILE *fpoint;
7      fpoint=fopen(IPTxtfname,"r");
8      while(fpoint==NULL){
9          printf("FAULT\n");
10         printf("Input File Name:\n");
11         scanf("%s\n", &IPTxtfname);
12         fpoint = fopen(IPTxtfname, "r");
13     }
14     char buffer[350];
15     fgets(buffer,350,fpoint);
16     fscanf(fpoint,"%s\n",OPTxtfname);
17     fgets(buffer,350,fpoint);
18     Point_ptr->point_x=(int*)malloc(sizeof(int)*50);
19     Point_ptr->point_y=(int*)malloc(sizeof(int)*50);
20     int *X=(int*)Point_ptr->point_x;
21     int *Y=(int*)Point_ptr->point_y;
22
23     int i;
24     for(i=0;i<50;i++)
25         fscanf(fpoint,"%d,%d\n",&X[i],&Y[i]);
26     fclose(fpoint);
27 }

```

■ Line1Function.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4  #include "Headerfile.h"
5  void Line1Function(Point *Point_ptr,Line1 *Line1_ptr)
6  {
7      float d,m;
8      float Total_d;
9      float i;
10     int j;
11     int *X=(int*)Point_ptr->point_x;
12     int *Y=(int*)Point_ptr->point_y;
13     Line1_ptr->SumDis1=0;

```

```

15 for(i=1;i<=20;i++){
16     Total_d=0;
17     m=pow(2,i);
18     for(j=0;j<50;j++){
19         d=fabs(-m*(float)X[j]+(float)Y[j])/pow(pow(m,2.0)+1,0.5);
20         Total_d+=d;
21     }
22
23     if((Line1_ptr->SumDis1)==0||Total_d<(Line1_ptr->SumDis1)){
24         Line1_ptr->SumDis1=Total_d;
25         Line1_ptr->Line1_m=m;
26     }
27 }
28 }

```

■ Line2Function.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4  #include "Headerfile.h"
5  void Line2Function(Point *Point_ptr,Line2 *Line2_ptr)
6  {
7      float Diff_Xi_Xavg,Diff_Yi_Yavg;
8      float Sum_MultiDxDy=0,Sum_DxSqua=0;
9      float Xavg,Yavg,Sum_X=0,Sum_Y=0;
10     int i;
11     float temp;
12     int *X=(int*)Point_ptr->point_x;
13     int *Y=(int*)Point_ptr->point_y;
14
15     for(i=0;i<50;i++){
16         Sum_X+=(float)X[i];
17         Sum_Y+=(float)Y[i];
18     }
19     Xavg=Sum_X/50;
20     Yavg=Sum_Y/50;
21
22     for(i=0;i<50;i++){
23         Diff_Xi_Xavg=(float)X[i]-Xavg;
24         Diff_Yi_Yavg=(float)Y[i]-Yavg;
25         temp=Diff_Xi_Xavg*Diff_Yi_Yavg;
26         Sum_MultiDxDy+=temp;
27         temp=pow(Diff_Xi_Xavg,2);
28         Sum_DxSqua+=temp;
29     }
30     Line2_ptr->Line2_m= Sum_MultiDxDy/Sum_DxSqua;
31     Line2_ptr->Line2_b=Yavg-(Line2_ptr->Line2_m)*Xavg;
32
33     float y_line;
34     Line2_ptr->SumDis2=0;
35     for(i=0;i<50;i++){
36         y_line=(Line2_ptr->Line2_m)*(float)X[i]+(Line2_ptr->Line2_b);
37         temp=fabs((float)Y[i]-y_line);
38         Line2_ptr->SumDis2+=temp;
39     }
40 }

```

■ WriteResult.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "Headerfile.h"
4  void WriteResult(char OPTxtfname[], Line1 *Line1_ptr, Line2 *Line2_ptr)
5  {
6      FILE *fwrite;
7      fwrite=fopen(OPTxtfname, "w");
8      while(fwrite==NULL){
9          printf("FAULT\n");
10         printf("Output File Name:\n");
11         scanf("%s\n", &OPTxtfname);
12         fwrite = fopen(OPTxtfname, "w");
13     }
14     float Diff;
15     fprintf(fwrite, "=====輸出結果=====\n");
16     Diff=(Line2_ptr->SumDis2)-(Line1_ptr->SumDis1);

17     fprintf(fwrite, "Line1: y = %.2fx\n", Line1_ptr->Line1_m);
18     fprintf(fwrite, "SumDis1 = %.2f\n", Line1_ptr->SumDis1);
19     fprintf(fwrite, "Line2: y = %.2fx + %.2f\n",
20     Line2_ptr->Line2_m, Line2_ptr->Line2_b);
21     fprintf(fwrite, "SumDis2 = %.2f\n", Line2_ptr->SumDis2);
22     fprintf(fwrite, "Diff = SumDis2 - SumDis1\n      = %.2f - %.2f = %.2f",
23     Line2_ptr->SumDis2, Line1_ptr->SumDis1, Diff);
24     printf("Diff = SumDis2 - SumDis1\n      = %.2f - %.2f = %.2f",
25     Line2_ptr->SumDis2, Line1_ptr->SumDis1, Diff);
26
27     fclose(fwrite);
28 }

```

□ 輸入文字檔 InputPoint.txt

InputPoint.txt - 記事本

檔案(F) 編輯(E) 格式(O) 檢視

--輸出運算結果文字檔
OutputResult.txt
--輸入50個座標點:

(1,18)	(11,88)	(21,158)	(31,228)	(41,298)
(2,17)	(12,87)	(22,157)	(32,227)	(42,297)
(3,32)	(13,102)	(23,172)	(33,242)	(43,312)
(4,31)	(14,101)	(24,171)	(34,241)	(44,311)
(5,46)	(15,116)	(25,186)	(35,256)	(45,326)
(6,45)	(16,115)	(26,185)	(36,255)	(46,325)
(7,60)	(17,130)	(27,200)	(37,270)	(47,340)
(8,59)	(18,129)	(28,199)	(38,269)	(48,339)
(9,70)	(19,140)	(29,210)	(39,280)	(49,350)
(10,76)	(20,146)	(30,216)	(40,286)	(50,356)

□ 執行結果

```
C:\Users\Jessie\Dropbox\嵌入式程式作業\期中報告迴歸直線_ver.2\Linear Regression .exe
--輸入座標點文字檔: InputPoint.txt
--輸出運算結果文字檔: OutputResult.txt
--輸入50個座標點:
( 1 , 18 ) ( 2 , 17 ) ( 3 , 32 ) ( 4 , 31 ) ( 5 , 46 )
( 6 , 45 ) ( 7 , 60 ) ( 8 , 59 ) ( 9 , 70 ) ( 10 , 76 )
( 11 , 88 ) ( 12 , 87 ) ( 13 , 102 ) ( 14 , 101 ) ( 15 , 116 )
( 16 , 115 ) ( 17 , 130 ) ( 18 , 129 ) ( 19 , 140 ) ( 20 , 146 )
( 21 , 158 ) ( 22 , 157 ) ( 23 , 172 ) ( 24 , 171 ) ( 25 , 186 )
( 26 , 185 ) ( 27 , 200 ) ( 28 , 199 ) ( 29 , 210 ) ( 30 , 216 )
( 31 , 228 ) ( 32 , 227 ) ( 33 , 242 ) ( 34 , 241 ) ( 35 , 256 )
( 36 , 255 ) ( 37 , 270 ) ( 38 , 269 ) ( 39 , 280 ) ( 40 , 286 )
( 41 , 298 ) ( 42 , 297 ) ( 43 , 312 ) ( 44 , 311 ) ( 45 , 326 )
( 46 , 325 ) ( 47 , 340 ) ( 48 , 339 ) ( 49 , 350 ) ( 50 , 356 )
Line1: y=8.0x
SumDis1 = 122.55
Line2: y=7.0x+7.2
SumDis2 = 164.88
Diff = SumDis2 - SumDis1
      = 164.88 - 122.55 = 42.33
-----
Process exited after 0.6354 seconds with return value 0
請按任意鍵繼續 . . .
```

□ 輸出文字檔 OutputResult.txt

```
OutputResult.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
=====輸出結果=====
Line1: y = 8.00x
SumDis1 = 122.55
Line2: y = 6.99x + 7.15
SumDis2 = 164.88
Diff = SumDis2 - SumDis1
      = 164.88 - 122.55 = 42.33
```