

# 中華民國第 53 屆中小學科學展覽會

## 作品說明書

---

國中組 物理科

佳作

030101

兩平面鏡任意夾角成像公式推導

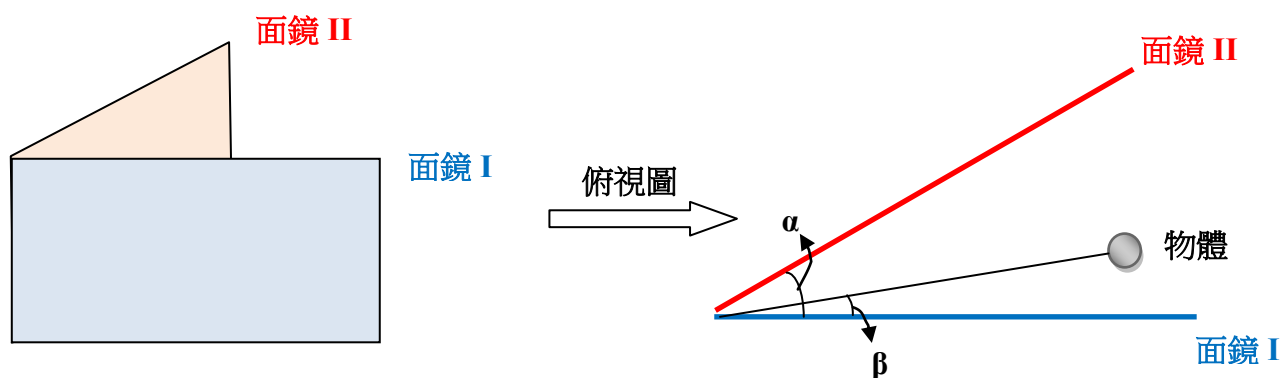
學校名稱：新北市私立光仁高級中學附設國中部

作者：  國二 吳易修	指導老師：  吳志堅
-------------------	------------------

關鍵詞：多次反射成像

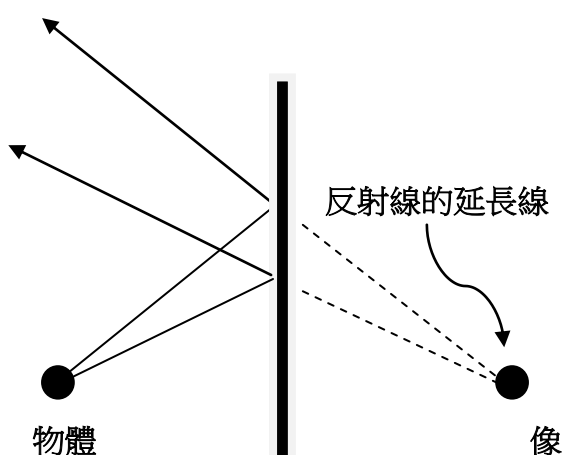
## 作品名稱：兩平面鏡任意夾角成像數公式推導

**摘要：**若有兩平面鏡( 面鏡 I、面鏡 II )，兩面鏡間夾角為  $\alpha$ ，物體與兩面鏡交點連線段與面鏡 I 的夾角為  $\beta$  ( $\alpha$ 、 $\beta$  單位為「度」)。在已知  $\alpha$ 、 $\beta$  的情況下，算出所能形成的最多像數，利用數學輔以所自行撰寫的 C 語言程式推導公式。

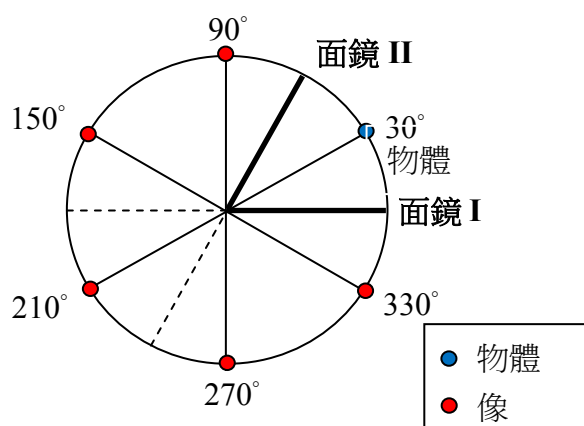


### 壹、研究動機

學習國中理化時，有提到反射定律和平面鏡成像的原理。在只有一面鏡子時，一個物體只會在鏡後形成一個像，當我們把兩面鏡子合在一起時，形成的像便不只一個。在書上或網路上查詢相關資料，對於「多次反射成像」這個部分，只有「經驗公式」能夠參考，當我在使用此公式時，發現在  $\frac{360}{\alpha}$  不為正整數時，會與實際結果不符，於是興起了推導公式的想法。



只有一面鏡子時



有兩面鏡子時( $\alpha=60^\circ$ ， $\beta=30^\circ$ )

## 貳、 研究目的

希望利用自行撰寫的 C 語言程式，觀察總像數隨著兩面鏡間夾角( $\alpha$ )、物體與兩面鏡交點連線段與面鏡 I 的夾角( $\beta$ )變化的情形，藉此推導出公式。

## 參、 文獻探討

經驗公式：由經驗可得多數情形符合此公式，公式如下：

一、  $\frac{360}{\alpha}$  為偶數：像數  $= \frac{360}{\alpha} - 1$

二、  $\frac{360}{\alpha}$  為奇數： 
$$\begin{cases} \text{物體在分角線上：像數} = \frac{360}{\alpha} - 1 \\ \text{物體不在分角線上：像數} = \frac{360}{\alpha} \end{cases}$$

三、  $\frac{360}{\alpha}$  不為正整數：像數  $= \left[ \frac{360}{\alpha} \right]$  ( $[ ]$  為高斯符號)

## 肆、 研究設備器材

電腦、紙、筆、圓規、量角器。

## 伍、 研究過程及方法

- 一、 由於實際上以兩面鏡子作實驗時，如果鏡子所夾角度( $\alpha$ )太小，很難數出總像數，有操作上的困難，因此選擇以程式語言來取代實際實驗，省時且較為精確。
- 二、 首先撰寫了程式一，能在輸入  $\alpha$ 、 $\beta$  兩個變數後，於螢幕輸出每個像的位置，並計算出總像數。
- 三、 以程式一檢查經驗公式中的第一點( $\frac{360}{\alpha}$  為偶數時)，證實了其總像數  $= \frac{360}{\alpha} - 1$ ，且觀察到最後一次的成像都會落在同一點。
- 四、 為節省更多時間，撰寫了程式二，在輸入  $\alpha$  後，使  $\beta$  從 0 度開始，每次增加 0.0625 度，並分別計算總像數，直到  $\beta = \alpha$  時停止。例如  $\alpha = 8^\circ$  時，經程式二執行後，結果如下表：

$\beta$ 的範圍(度)	0.0625 ~ 3.9375	4.0000	4.0625 ~ 7.9375
總像數(個)	45	44	45

五、 以程式二檢查經驗公式中的第二點( $\frac{360}{\alpha}$ 為奇數時)，並且證實其無誤。

六、 開始檢查經驗公式中的第三點( $\frac{360}{\alpha}$ 不為正整數時)，用程式二檢驗時，發現輸出來的總像數會隨著  $\beta$  變大，而有特殊的變化。

七、 針對 $\frac{360}{\alpha}$ 不為正整數時，發現總像數的變化方式有兩種，舉例說明如下：

第一種(如： $\alpha = 7^\circ$ ) $\beta$ 的範圍(度)	0.0625 ~ <b>2.0000</b>	2.0625 ~ 4.9375	<b>5.0000</b> ~ 6.9375
總像數(個)	51	52	51
第二種(如： $\alpha = 11^\circ$ ) $\beta$ 的範圍(度)	0.0625 ~ 3.9375	<b>4.0000</b> ~ <b>7.0000</b>	7.0625 ~ 10.9375
總像數(個)	33	32	33

八、 在此，我稱第一種總像數的變化方式為「小—大—小」(如：51—52—51)，第二種變化方式為「大—小—大」(如：33—32—33)。

九、 觀察上面的表格，我發現在  $\alpha$  為正整數時， $\beta$  會有兩個整數臨界值(紅字部分)，且兩個臨界值將  $\beta$  變化的範圍分成三段，在此我稱第一個臨界值為 $\beta_1$ ，第二個為 $\beta_2$  ( $\beta_1$ 、 $\beta_2$ 單位為「度」)。

十、 而經驗公式中所提到的，當 $\frac{360}{\alpha}$ 不為正整數時，像數 =  $\left\lceil \frac{360}{\alpha} \right\rceil$ ，但從第七點的表格來看，不論是「小—大—小」還是「大—小—大」，符合經驗公式的都是「小」的部分，而大的部分總像數應為  $\left\lceil \frac{360}{\alpha} \right\rceil + 1$  (個)。

十一、 因此將上方表格整理如下：

「小一大一小」 $\beta$ 的範圍(度)	$0 < \beta \leq \beta_1$	$\beta_1 < \beta < \beta_2$	$\beta_2 \leq \beta < \alpha$
總像數(個)	$\left\lceil \frac{360}{\alpha} \right\rceil$	$\left\lceil \frac{360}{\alpha} \right\rceil + 1$	$\left\lceil \frac{360}{\alpha} \right\rceil$
「大一小一大」 $\beta$ 的範圍(度)	$0 < \beta < \beta_1$	$\beta_1 \leq \beta \leq \beta_2$	$\beta_2 < \beta < \alpha$
總像數(個)	$\left\lceil \frac{360}{\alpha} \right\rceil + 1$	$\left\lceil \frac{360}{\alpha} \right\rceil$	$\left\lceil \frac{360}{\alpha} \right\rceil + 1$

十二、我程式二繼續檢查，證實了 $\frac{360}{\alpha}$ 不為正整數時，總像數隨著 $\beta$ 的變化只有以上

兩種，而意識到，只要能找出 $\beta_1$ 、 $\beta_2$ 與 $\alpha$ 之間的關係，便能推導出公式。

十三、由於一個 $\alpha$ 只有對應一組 $\beta_1$ 和 $\beta_2$ ，便開始觀察三者間的關係，卻只有看出

$\alpha = \beta_1 + \beta_2$ ，所以我花了一點時間，把數據都輸進試算表(90°以內)，繼續找尋關係。

小-大-小	( $\alpha$ )	$\beta_1$	$\beta_2$		大-小-大	( $\alpha$ )	$\beta_1$	$\beta_2$	( $\alpha$ )	$\beta_1$	$\beta_2$
7		2	5		11		4	7	81	18	63
13		2	11		16		4	12	82	16	66
14		2	12		19		9	10	83	14	69
17		7	10		22		4	18	84	12	72
21		9	12		25		5	20	85	10	75
23		4	19		28		12	16	86	8	78
26		2	24		29		6	23	87	6	81
27		9	18		33		15	18	88	4	84
31		6	25		34		10	24	89	2	87
32		12	20		35		5	30			
37		5	32		41		16	25			
38		10	28		42		12	30			
39		15	24		43		8	35			
46		4	42		44		4	40			
47		8	39		52		24	28			
48		12	36		53		21	32			
49		16	33		54		18	36			
50		20	30		55		15	40			
51		24	27		56		12	44			
61		3	58		57		9	48			
62		6	56		58		6	52			
63		9	54		59		3	56			
64		12	52		73		34	39			
65		15	50		74		32	42			
66		18	48		75		30	45			
67		21	46		76		28	48			
68		24	44		77		26	51			
69		27	42		78		24	54			
70		30	40		79		22	57			
71		33	38		80		20	60			

十四、我找了很久，想了很多種可能，但是都不對。偶然間，想到可能會跟餘數有關係，於是拿 180 當被除數， $\alpha$  當除數，結果發現得到的餘數恰好就是「小一大一小」中的  $\beta_2$ ，以及「大一小一大」中的  $\beta_1$ 。

十五、敘述中的餘數定義如下：當被除數除以除數時，在商為最大可能的整數時，被除數尚未被除盡的值。例如： $180 \div 23.5 = 7 \dots 15.5$ ，因此最大可能的整數商數為 7，餘數即為 15.5。在 C 語言中代表餘數的運算子是「 $\%$ 」，因此在以下敘述中會使用此符號，當算式為  $17.9 \div 2.3 = 7 \dots 1.8$  時，會寫成  $17.9 \% 2.3 = 1.8$ 。

十六、根據第十三點提到的  $\alpha = \beta_1 + \beta_2$ ，和第十四點發現的關係，得到以下表格：

像數變化為「小一大一小」時	$\alpha - 180 \% \alpha = \beta_1$	$180 \% \alpha = \beta_2$
像數變化為「大一小一大」時	$180 \% \alpha = \beta_1$	$\alpha - 180 \% \alpha = \beta_2$

十七、於是現在只需要找到一個辦法，可以在知道  $\alpha$  後，判斷總像數的變化方式是「小一大一小」還是「大一小一大」。

十八、所以我繼續使用試算表，最後終於發現了其中的關係。當  $180 \div \alpha$  時，如果把『商去掉整數位』後所剩的值大於 0.5，則總像數的變化方式為「小一大一小」；小於 0.5 但大於 0 時則為「大一小一大」；等於 0.5 時，就是  $\frac{360}{\alpha}$  為奇數；等於 0 時，就是  $\frac{360}{\alpha}$  為偶數時。於是只要算出  $(180 \div \alpha) \% 1$  的答案，就可以利用這個值來判斷  $\alpha$  屬於哪一種類型(如下表)：

	類型	$\frac{180}{\alpha}$ 的商去掉整數位後的數值
第一種(如 $\alpha = 30^\circ$ )	$\frac{360}{\alpha}$ 為偶數	等於 0 (6.0)
第二種(如 $\alpha = 40^\circ$ )	$\frac{360}{\alpha}$ 為奇數	等於 0.5 (4.5)
第三種(如 $\alpha = 50^\circ$ )	像數變化為 小一大一小	大於 0.5 (3.6)
第四種(如 $\alpha = 75^\circ$ )	像數變化為 大一小一大	小於 0.5 且不等於 0 (2.4)

十九、綜合第三、五、十一、十六、十八點，可以推導出公式，也是這次研究的主要目的及結論。

## 陸、 研究結果

一、 將  $\alpha$  在不同情況時的所有整數可能值，整理成表格如下：

	$\alpha$ 的所有整數可能值 (度)	
$\frac{360}{\alpha}$ 為偶數時	1、2、3、4、5、6、9、10、12、15、18、20、30、36、45、60、90、180	
$\frac{360}{\alpha}$ 為奇數時	8、24、40、72、120	
$\frac{360}{\alpha}$ 不為正整數時	像數變化為 小一大一小	7、13、14、17、21、23、26、27、31、32、37、38、39、46、47、48、49、50、51、61、62、63、64、65、66、67、68、69、70、71、91、92、93、94、95、96、97、98、99、100、101、102、103、104、105、106、107、108、109、110、111、112、113、114、115、116、117、118、119
	像數變化為 大一小一大	11、16、19、22、25、28、29、33、34、35、41、42、43、44、52、53、54、55、56、57、58、59、73、74、75、76、77、78、79、80、81、82、83、84、85、86、87、88、89、121、122、123、124、125、126、127、128、129、130、131、132、133、134、135、136、137、138、139、140、141、142、143、144、145、146、147、148、149、150、151、152、153、154、155、156、157、158、159、160、161、162、163、164、165、166、167、168、169、170、171、172、173、174、175、176、177、178、179

二、 利用程式一，列出所有符合  $\frac{360}{\alpha}$  為偶數的  $\alpha$ ，及其對應的像數：

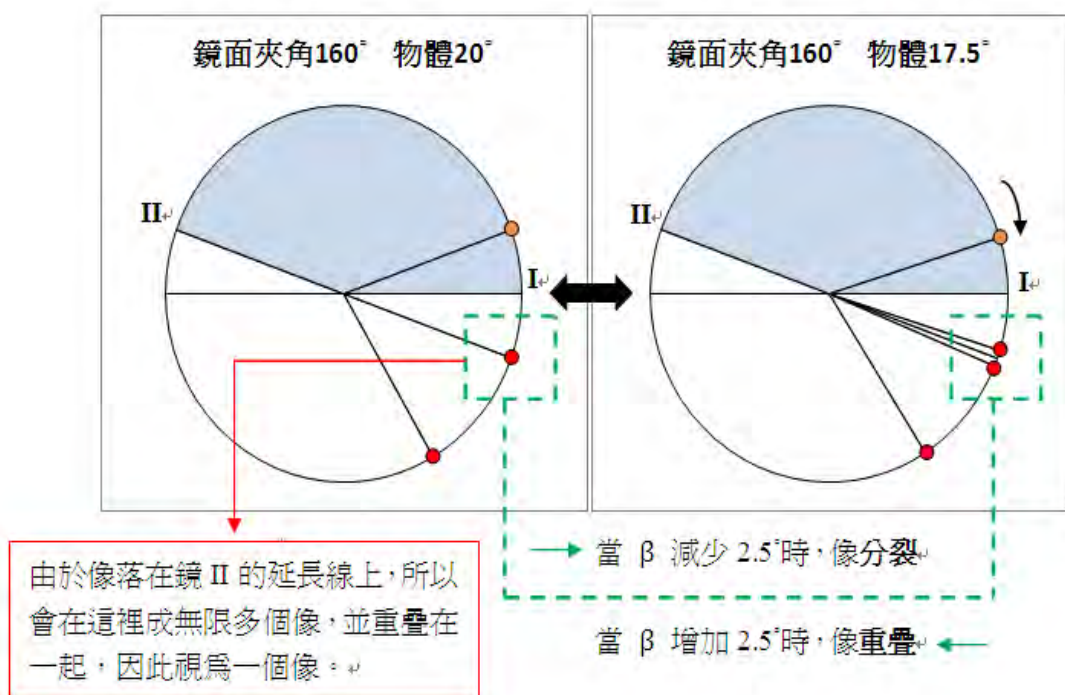
$\alpha$ (度)	1	2	3	4	5	6	9	10	12	15	18	20	30	36	45	60	90	180
像數(個)	359	179	119	89	71	59	39	35	29	23	19	17	11	9	7	5	3	1

三、 利用程式一，列出所有符合  $\frac{360}{\alpha}$  為奇數的  $\alpha$ ，及其對應的像數：

$\alpha$ (度)		8	24	40	72	120
像數(個)	$\beta \neq \frac{\alpha}{2}$	45	15	9	5	3
	$\beta = \frac{\alpha}{2}$	44	14	8	4	2

## 柒、 討論

在此深入探討  $\frac{360}{\alpha}$  不為正整數時，總像數會隨著  $\beta$  變化的原因。經由先前整理出來的結果，可以知道  $\beta$  從 0 開始慢慢增大的過程中，會在  $\beta = \beta_1$  或  $\beta = \beta_2$  時變化像數，有時會多一個，有時則是少一個。至於會多一個或少一個的原因，是因為最後一個像的「分裂」及「重疊」，而發生分裂或重疊的時機，便是物體在  $\beta_1$  或是  $\beta_2$  的時候，今以  $\alpha = 160^\circ$  ( $\beta_1 = 20^\circ$ ) 做為例子說明。





## 捌、 結論

一、 若有兩平面鏡夾角為  $\alpha$ ，物體在  $\beta$ ，則此物體可形成的最多像數公式如下：

(%為計算餘數的符號；[]為高斯符號)

先計算  $180 \div \alpha$  去掉整數位後所得之數  $\rightarrow (180 \div \alpha) \% 1$

(一) 去整數位後為 0：

$$\text{像數} = \frac{360}{\alpha} - 1$$

(二) 去整數位後為 0.5：

$$\text{若 } \beta = \frac{\alpha}{2}, \text{ 像數} = \frac{360}{\alpha} - 1$$

$$\text{若 } \beta \neq \frac{\alpha}{2}, \text{ 像數} = \frac{360}{\alpha}$$

(三) 去整數位後  $> 0.5$ ：

$$180 \% \alpha = \beta_2, \alpha - \beta_2 = \beta_1$$

$$\text{若 } \beta_1 < \beta < \beta_2, \text{ 像數} = \left[ \frac{360}{\alpha} \right] + 1$$

$$\text{若 } \beta \leq \beta_1 \text{ or } \beta \geq \beta_2, \text{ 像數} = \left[ \frac{360}{\alpha} \right]$$

(四) 去整數位後  $< 0.5$  且  $\neq 0$ ：

$$180 \% \alpha = \beta_1, \alpha - \beta_1 = \beta_2$$

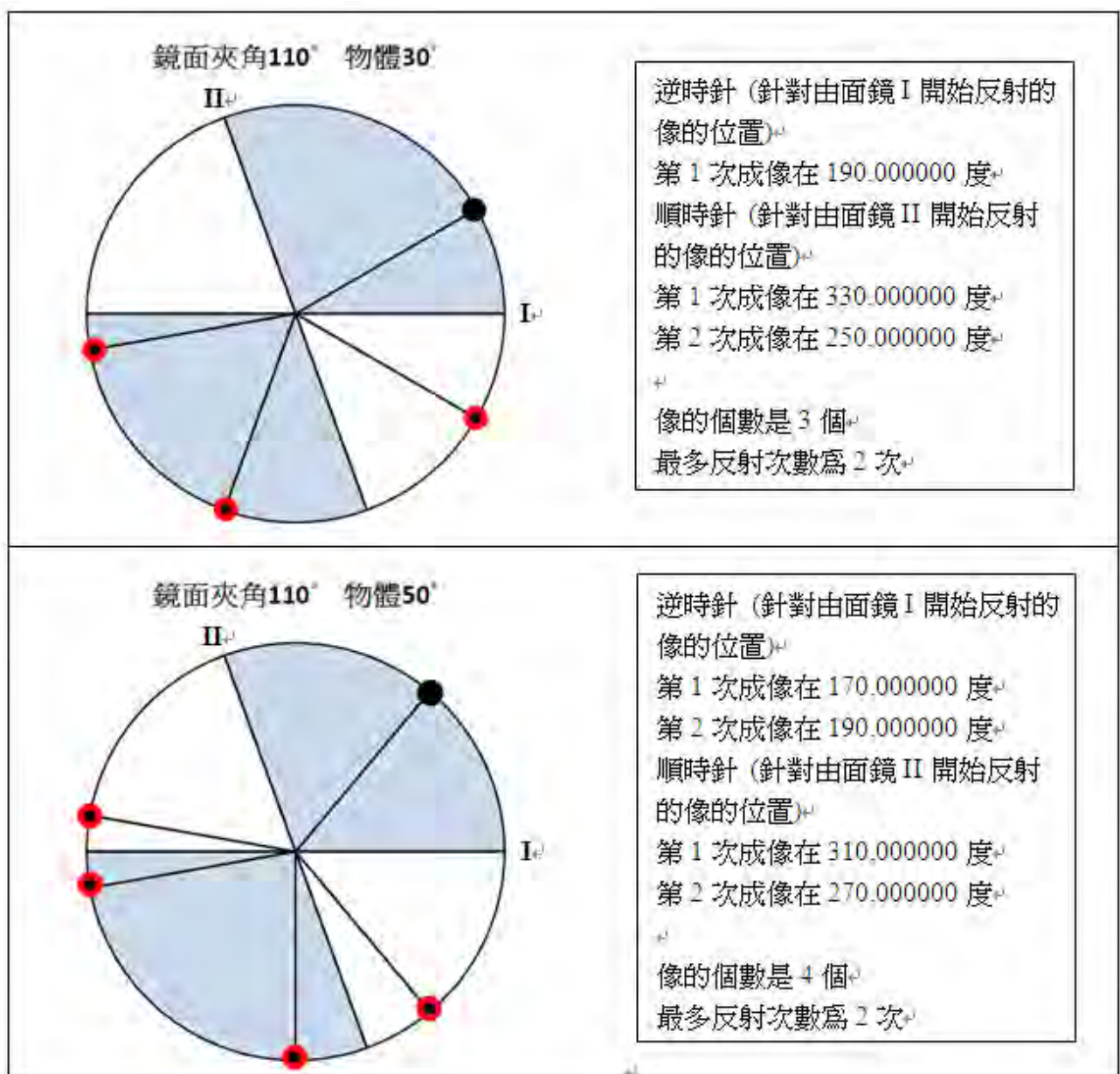
$$\text{若 } \beta_1 \leq \beta \leq \beta_2, \text{ 像數} = \left[ \frac{360}{\alpha} \right]$$

$$\text{若 } \beta < \beta_1 \text{ or } \beta > \beta_2, \text{ 像數} = \left[ \frac{360}{\alpha} \right] + 1$$

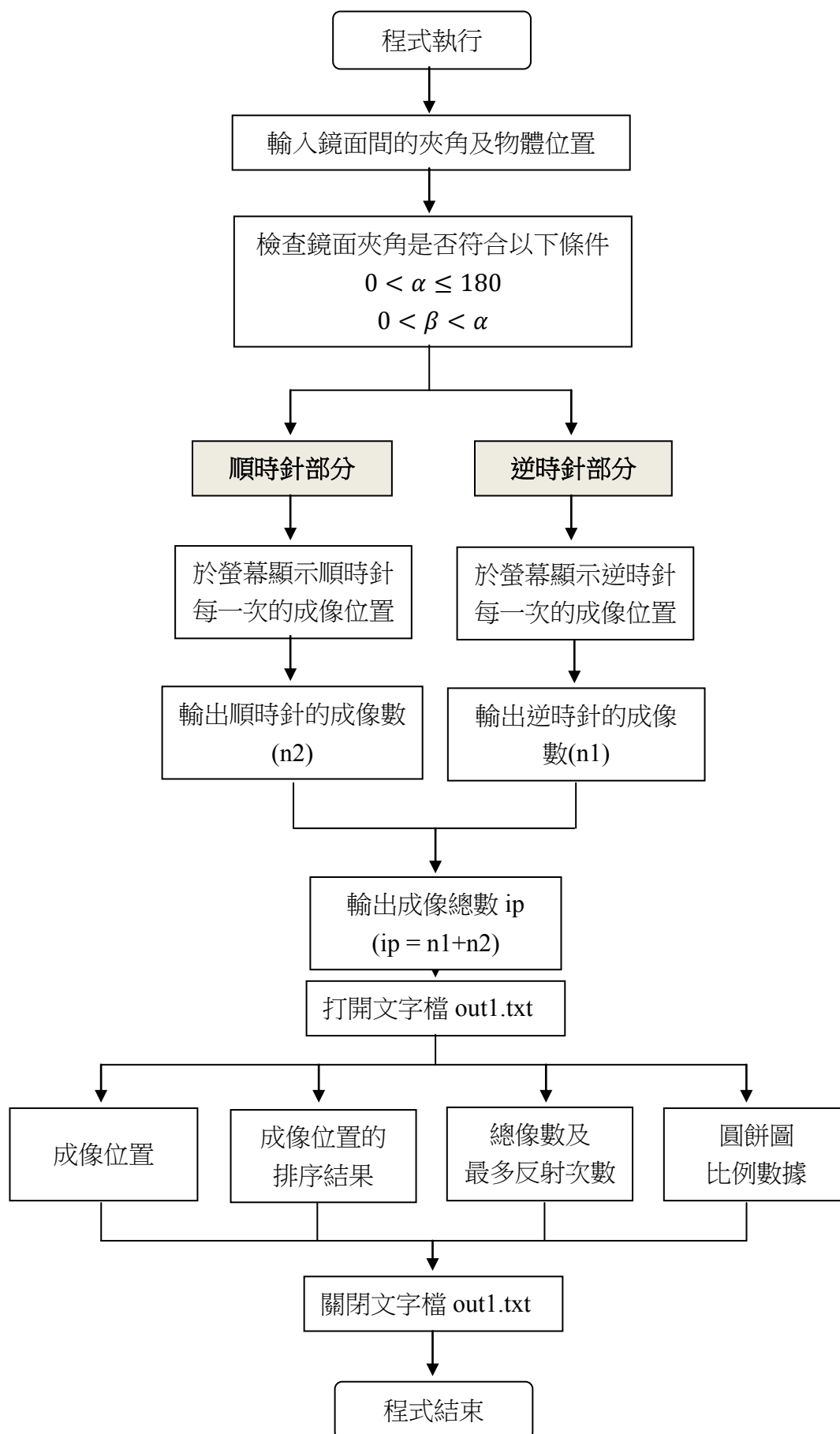
二、 針對結論，寫了最後一個程式程式三，功能就是在輸入鏡子夾角及物體角度後，在螢幕輸出每個像的成像位置，以及總像數，並且將數據轉成文字檔，並加以歸納統整。程式三的總輸出內容如下：

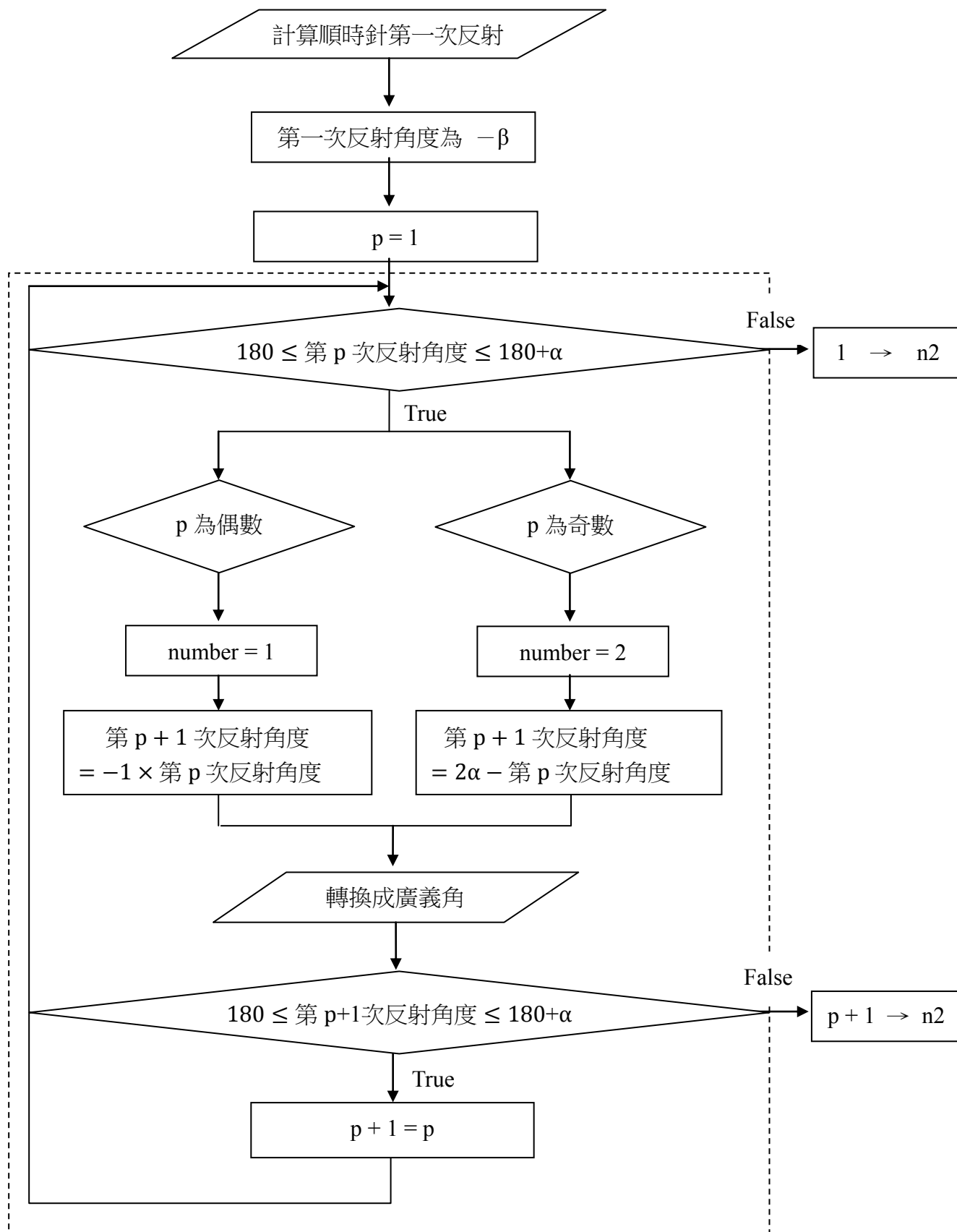
螢幕	成像位置		總像數	
文字檔	成像位置	成像位置的排序結果	總像數及最多反射次數	圓餅圖比例數據

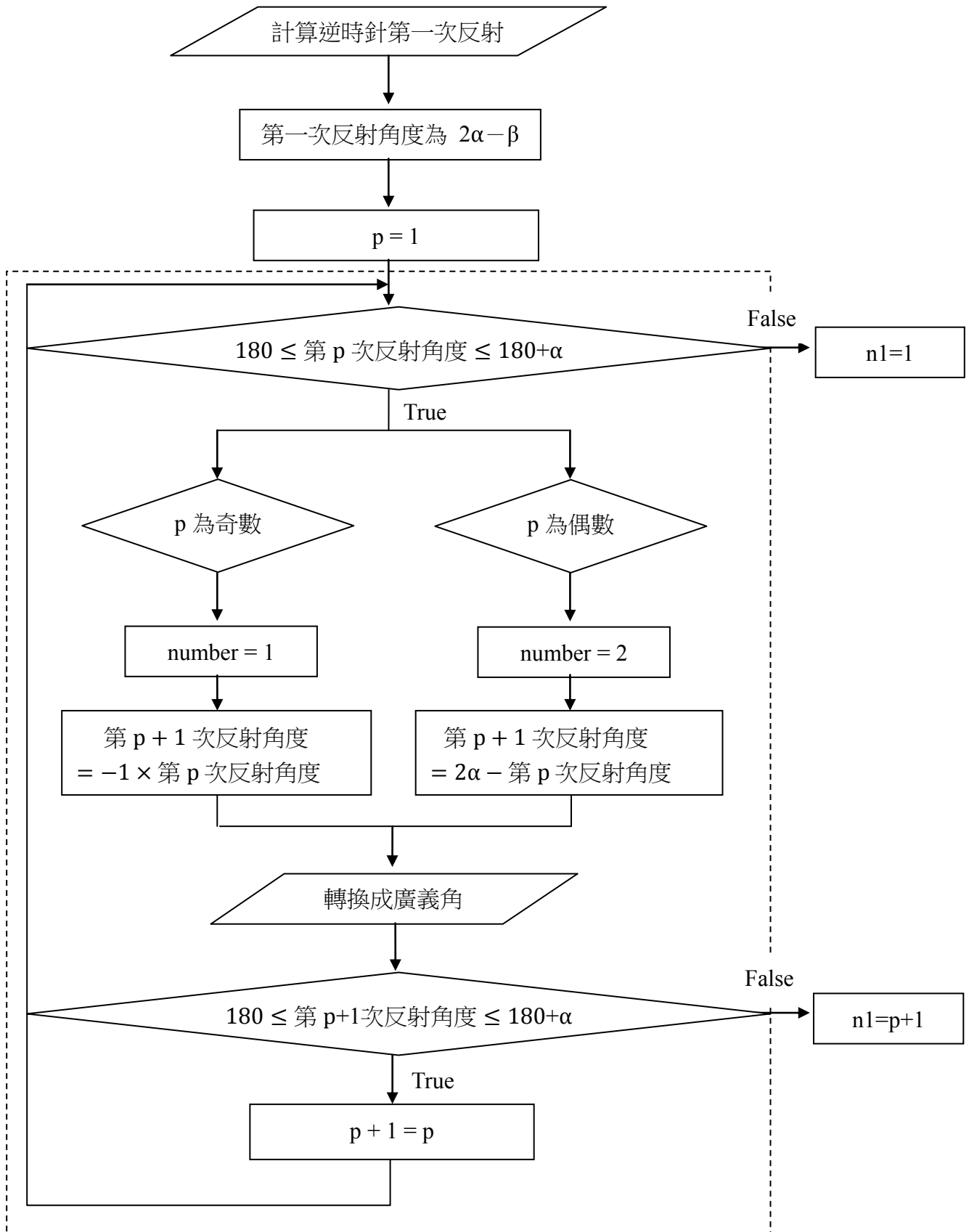
三、 用圓餅圖畫出的成像圖



四、 以下繪出了程式三的程式流程圖：







玖、 參考資料及其他(教科書無標示版次)

- 一、 P84~P88，自然與生活科技 2 上/翰林出版
- 二、 易傳傑、王一哲/活用高三選修物理上 A/建宏出版社
- 三、 姚珩/普通高級中學基礎物理(一)/翰林出版

## 程式

程式一                      \*在程式中的順、逆時針是指第一次反射的方向\*

程式碼如下

---

```
#include<stdio.h>
#include<conio.h>
#define num1 100000
#define num2 100000

double reflection(double num);/*計算反射後角度*/
double turn(double num);/*轉換廣義角*/
double a;/*鏡面夾角*/
int b;/*像的數量*/
double c;/*物體與零度鏡面夾角*/
double cw[num1];/*順時針方向*/
double ccw[num2];/*逆時針方向*/
int number;
int j, i, p, n1, n2;

int main(void)
{
    /*判斷輸入值*/
    for(j=0;j++){
        printf("請輸入鏡面夾角:\n");
        scanf("%lf", &a);
        if(a<=0 || a>180){
            printf("鏡面夾角不能小於等於零或大於一百八十度\n");
            continue;}
        else {}
        printf("請輸入物體與零度鏡面夾角:\n");
        scanf("%lf", &c);
        if(c>=a||c<=0){
            printf("物體與零度鏡面夾角不能大於等於鏡面夾角或小於等於零\n");
        }
        else {break;}
    }

    /*第一次反射*/
    cw[0] = turn(-c);
    ccw[0] = turn(2*a-c);
    /*主程式*/
    for(p=1;;p++){
        if(180<=ccw[p-1] && ccw[p-1]<=180+a){
            n1=1;
            break;}
```

```

        if(p%2==1){
            number = 1;}
        else{
            number = 2;}
        ccw[p] = reflection(ccw[p-1]);
        ccw[p] = turn(ccw[p]);

        if(180<=ccw[p] && ccw[p]<=180+a){
            n1=p+1;
            break;}
    }

for(p=1;;p++){
    if(180<=cw[p-1] && cw[p-1]<=180+a){
        n2=1;
        break;}

    if(p%2==1){
        number = 2;}
    else{
        number = 1;}
    cw[p] = reflection(cw[p-1]);
    cw[p] = turn(cw[p]);

    if(180<=cw[p] && cw[p]<=180+a){
        n2=p+1;
        break;}
}

for(i=1;;i++){
    if(ccw[i-1]==0){
        printf("像的個數是%d 個\n\n", n1);
        break;}
    else{
        printf("逆時針的第%d 次成像在%f 度\n", i, ccw[i-1]);}
}

for(i=1;;i++){
    if(cw[i-1]==0){
        printf("像的個數是%d 個\n\n", n2);
        break;}
    else{

```



```

        printf("順時針的第%d 次成像在%f 度\n", i, cw[i-1]);}
    }

    if(ccw[n1-1]!=cw[n2-1])/*檢查是否重複*/{
        printf("像的總個數為%d 個", n1+n2);}
    else{
        printf("像的總個數為%d 個", n1+n2-1);}

    getch();
}
double reflection(double num)
/*計算反射後角度*/
{
    switch(number){
        case 1:
            return -num;
            break;
        case 2:
            return 2*a-num;
            break;
    }
}
double turn(double num)
/*轉換廣義角*/
{
    if(num >=360){
        num -= 360;}
    else if(num < 0){
        num += 360;}
    else{
        num = num;}
    return num;
}

```

---

程式二

程式碼如下

---

```
#include<stdio.h>
#include<conio.h>
#define num1 100000
#define num2 100000

double reflection(double num);/*計算反射後角度*/
double turn(double num);/*轉換廣義角*/
int a; /*鏡面夾角*/
int b; /*像的數量*/
double c; /*物體與零度鏡面夾角*/
double cw[num1]; /*順時針方向*/
double ccw[num2]; /*逆時針方向*/
int number;
int i, p, n, n1, n2;

int main(void)
{
    printf("輸入鏡子夾角:\n");
    scanf("%d", &a);
    /*物體角度累加*/
    for(c=0.0625;c<a;c+=0.0625){

        /*第一次反射*/
        cw[0] = turn(-c);
        ccw[0] = turn(2*a-c);

        /*主程式*/
        for(p=1;;p++){
            if(180<=ccw[p-1] && ccw[p-1]<=180+a){
                n1=1;
                break;}

            if(p%2==1){
                number = 1;}
            else{
                number = 2;}
            ccw[p] = reflection(ccw[p-1]);
```

```

ccw[p] = turn(ccw[p]);

if(180<=ccw[p] && ccw[p]<=180+a){
    n1=p+1;
    break;}
}

for(p=1;;p++){
    if(180<=cw[p-1] && cw[p-1]<=180+a){
        n2=1;
        break;}

    if(p%2==1){
        number = 2;}
    else{
        number = 1;}
    cw[p] = reflection(cw[p-1]);
    cw[p] = turn(cw[p]);

    if(180<=cw[p] && cw[p]<=180+a){
        n2=p+1;
        break;}
}

for(i=1;;i++){
    if(ccw[i-1]==0){
        break;}
    else{}
}

for(i=1;;i++){
    if(cw[i-1]==0){
        break;}
    else{}
}

```

```

        if(ccw[n1-1]!=cw[n2-1]){
                                n=n1+n2;
                                printf("夾角為%f時,像的總個數為%d 個\n", c, n);}

        else{
                n=n1+n2-1;
                printf("夾角為%f時,像的總個數為%d 個\n", c, n);}
}

        getch();
}

double reflection(double num)
/*計算反射後角度*/
{
        switch(number){
        case 1:
                                return -num;
                                break;

        case 2:
                                return 2*a-num;
                                break;
                                }
}

double turn(double num)
/*轉換廣義角*/
{
        if(num >=360){
                num -= 360;}
        else if(num < 0){
                num += 360;}
        else{
                num = num;}
        return num;
}

```

---

### 程式三

程式碼如下

---

```
#include<stdio.h>
#include<conio.h>
#define num1 100000
#define num2 100000
#define num3 100000

FILE *fin, *fout;
double reflection(double num);/*計算反射後角度*/
double turn(double num);/*轉換廣義角*/
double a;/*鏡面夾角*/
int b;/*像的數量*/
double c;/*物體與零度鏡面夾角*/
double cw[num1];/*順時針方向*/
double ccw[num2];/*逆時針方向*/
double mix[num3];/*順+逆*/
int number;
int j, i, p, n1, n2;
int ii, ip;
int mixx;

int main(void)
{
    /*判斷輸入值*/
    for(j=0;;j++){
        printf("請輸入鏡面夾角:\n");
        scanf("%lf", &a);
        if(a<=0 || a>180){
            printf("鏡面夾角不能小於等於零或大於一百八十度\n");
            continue;}
        else {}
        printf("請輸入物體與零度鏡面夾角:\n");
        scanf("%lf", &c);
        if(c>=a||c<=0){
            printf("物體與零度鏡面夾角不能大於等於鏡面夾角或小於等於零\n");
        }
        else {break;}
    }
    /*第一次反射*/
```

```

cw[0] = turn(-c);
ccw[0] = turn(2*a-c);
/*主程式*/
for(p=1;;p++){
    if(180<=ccw[p-1] && ccw[p-1]<=180+a){
        n1=1;
        break;}

    if(p%2==1){
        number = 1;}
    else{
        number = 2;}
    ccw[p] = reflection(ccw[p-1]);
    ccw[p] = turn(ccw[p]);

    if(180<=ccw[p] && ccw[p]<=180+a){
        n1=p+1;
        break;}
}

for(p=1;;p++){
    if(180<=cw[p-1] && cw[p-1]<=180+a){
        n2=1;
        break;}

    if(p%2==1){
        number = 2;}
    else{
        number = 1;}
    cw[p] = reflection(cw[p-1]);
    cw[p] = turn(cw[p]);

    if(180<=cw[p] && cw[p]<=180+a){
        n2=p+1;
        break;}
}

```

```

for(i=1;;i++){
    if(ccw[i-1]==0){
        printf("像的個數是%d 個\n\n", n1);
        break;}
    else{
        printf("逆時針的第%d 次成像在%f 度\n", i, ccw[i-1]);}
}
for(i=1;;i++){
    if(cw[i-1]==0){
        printf("像的個數是%d 個\n\n", n2);
        break;}
    else{
        printf("順時針的第%d 次成像在%f 度\n", i, cw[i-1]);}
}

if(ccw[n1-1]!=cw[n2-1])/*檢查是否重複*/{
    printf("像的總個數為%d 個", n1+n2);
    ip=n1+n2;
    for(mixx=0;mixx<n2;mixx++){
        mix[mixx]=cw[mixx];
        mix[mixx+n2]=ccw[mixx];}
}
else{
    printf("像的總個數為%d 個", n1+n2-1);
    ip=n1+n2-1;
    for(mixx=0;mixx<n2-1;mixx++){
        mix[mixx]=cw[mixx];
    }
    for(mixx=0;mixx<n2;mixx++){
        mix[mixx+n2-1]=ccw[mixx];
    }
}

/*排序*/

double tmp;
int m, n , s, t;

for(s=0; s<ip-1; s++){
    for(t=s+1; t<ip; t++){

```

```

        if(mix[t] < mix[s]){
            tmp = mix[t];
            mix[t] = mix[s];
            mix[s] = tmp;
        }
    }
}

/*載入至 txt*/
FILE *fout;

fout=fopen("out1.txt","w+");
if(fout==NULL){
    printf("Fail To Open File out1.txt!!");
    return 0;}

fprintf(fout,"逆時針 (針對由面鏡 I 開始反射的像的位置)\n");
for(ii=0;ii<n1;ii++) {
    fprintf(fout,"第%d 次成像在",ii+1);
    fprintf(fout,"%f 度\n",ccw[ii]);}

fprintf(fout,"\n\n 順時針 (針對由面鏡 II 開始反射的像的位置)\n");
for(ii=0;ii<n2;ii++) {
    fprintf(fout,"第%d 次成像在",ii+1);
    fprintf(fout,"%f 度\n",cw[ii]);}

fprintf(fout,"\n\n 排序後結果\n");
for(ii=0;ii<ip;ii++) {
    fprintf(fout,"%f\n",mix[ii]);}

fprintf(fout,"\n\n 像的個數是%d 個\n", ip);

if(n1>=n2){
    fprintf(fout,"最多反射次數為%d 次\n\n\n", n1);}
else{
    fprintf(fout,"最多反射次數為%d 次\n\n\n", n2);}

/*圓餅圖比例數據*/
int q;

```



```

mix[ip]=0;
mix[ip+1]=a;
mix[ip+2]=c;
if(cw[n2-1]==180+a || ccw[n1-1]==180+a){
    mix[ip+3]=180;
    ip+=4;}
else if(cw[n2-1]==180 || ccw[n1-1]==180){
    mix[ip+3]=180+a;
    ip+=4;}
else{
    mix[ip+3]=180;
    mix[ip+4]=180+a;
    ip+=5;}

for(s=0; s<ip-1; s++){
    for(t=s+1; t<ip; t++){
        if(mix[t] < mix[s]){
            tmp = mix[t];
            mix[t] = mix[s];
            mix[s] = tmp;
        }
    }
}

fprintf(fout,"圓餅圖比例數據\n");
    fprintf(fout,"%f\n", 360-mix[ip-1]);
for(q=ip-1;q>0;q--){
    fprintf(fout,"%f\n", mix[q]-mix[q-1]);}

fclose(fout);

getch();
}

double reflection(double num)
/*計算反射後角度*/
{
    switch(number){
    case 1:
        return -num;

```

```

        break;
    case 2:
        return 2*a-num;
        break;
    }
}
double turn(double num)
/*轉換廣義角*/
{
    if(num >=360){
        num -= 360;}
    else if(num < 0){
        num += 360;}
    else{
        num = num;}
    return num;
}

```

## 【評語】 030101

1. 研究者很用心，進行資料與程式整理。
2. 可以在創意方面再加思考會更佳。
3. 應改慮成果之廣泛應用。