1. 請寫一程式，將 1.bmp 影像作水平鏡射和垂直鏡射。

Source Code：

```
#include <iostream>
#include "stdlib.h"
#include "bmp.h"
void horizontal(unsigned char **ima, unsigned char **bima, int nr,int nc);
void vertical(unsigned char **ima, unsigned char **bima, int nr,int nc);

using namespace std;

int main(int argc, char** argv) {

	unsigned char **ima, **bima;
	int nr,nc; //image height and width
	char filename[128],temp;
	bool isfilefine = false;

	//read bmp image from file
	cout << "Enter input filename:";
	cin >> filename;
	isfilefine = Read_BMP(filename, ima, nr, nc);
	if (!isfilefine)  return 0;
	bima=UC2D(nr,nc);
	Write_BMP_8bits("ima.bmp", ima, nr, nc);
	horizontal(ima,bima,nr,nc);
	Write_BMP_8bits("ex1_horizontal.bmp", bima, nr, nc);
	vertical(ima,bima,nr,nc);
	Write_BMP_8bits("ex1_vertical.bmp", bima, nr, nc);
	cout << "\nProgram done.\n";
	system("PAUSE");
	return 1;
}


void horizontal(unsigned char **ima, unsigned char **bima, int nr,int nc)
{
	for(int i=0;i<nr;i++)
		for(int j=nc-1;j>=0;j--)
			bima[i][nc-1-j]=ima[i][j];
}
```

```
void vertical(unsigned char **ima, unsigned char **bima, int nr,int nc)
{
	for(int i=nr-1;i>=0;i--)
		for(int j=0;j<nc;j++)
			bima[nr-1-i][j]=ima[i][j];
}
```

(a) (b) (c)

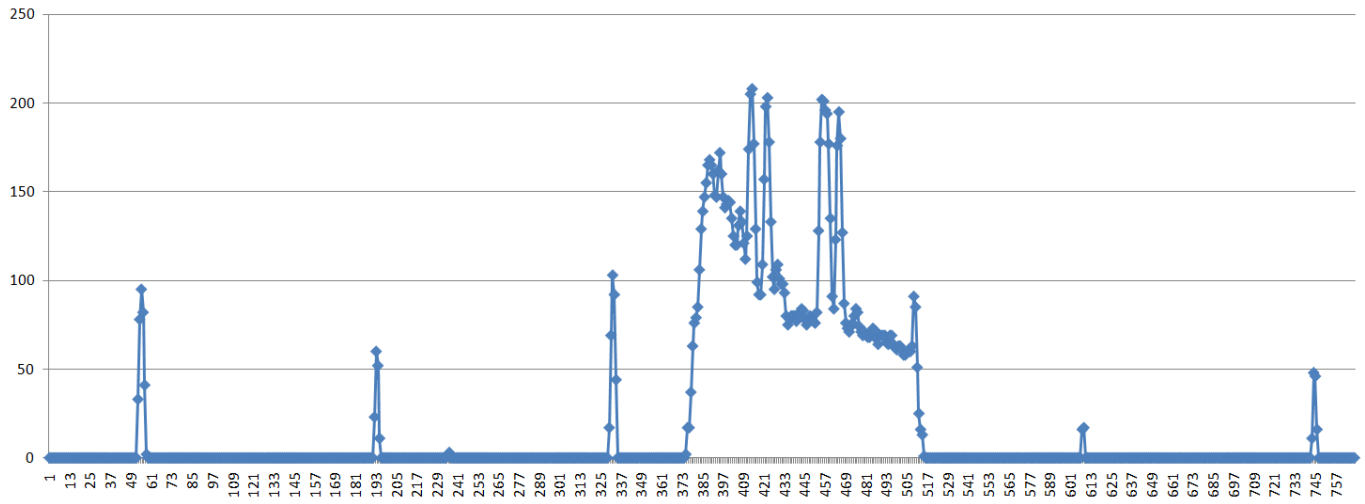
(a)原始圖檔 1.bmp、(b)水平鏡射 ex1_horizontal.bmp、(c)垂直鏡射 ex1_vertical.bmp

2. 將 1.bmp 第 100 row 的剖面資料取出，在 Excel 當中畫出其曲線圖。

Source Code：

```
#include <iostream>
#include "stdlib.h"
#include "bmp.h"
#include<fstream>
using namespace std;
void excel(unsigned char **ima,int nc);

using namespace std;

int main(int argc, char** argv) {
	unsigned char **ima, **bima;
	int nr,nc; //image height and width
	char filename[128],temp;
	bool isfilefine = false;

	//read bmp image from file
	cout << "Enter input filename:";
	cin >> filename;
	isfilefine = Read_BMP(filename, ima, nr, nc);
	if (!isfilefine)  return 0;
	excel(ima,nc);
```

```
        cout << "\nProgram done.\n";
        system("PAUSE");
        return 1;
}

void excel(unsigned char **ima,int nc)
{
        ofstream out("ex2_100row.xls");
        for(int j=0;j<nc;j++)
                out<<(int)ima[99][j]<<endl;
}
```



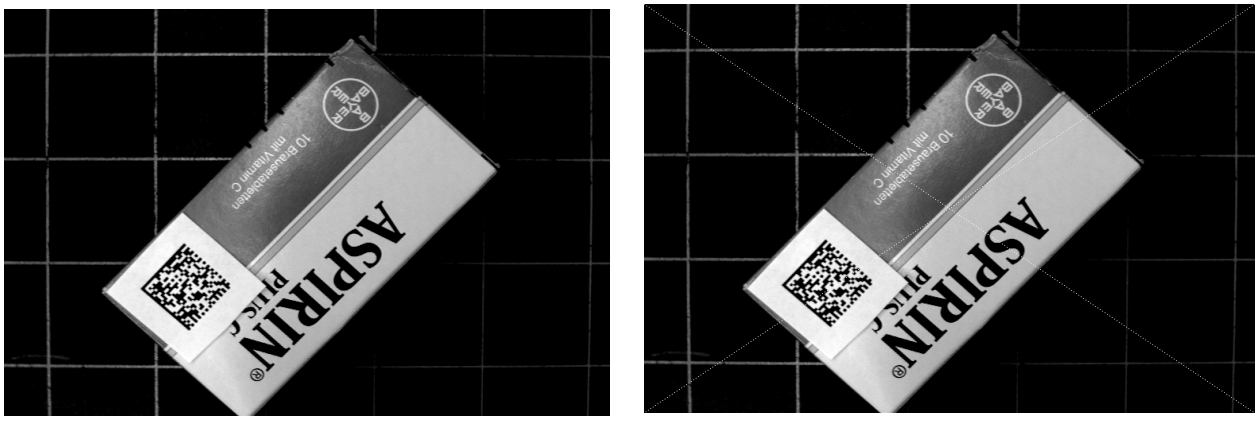
1.bmp 第 100 row 的剖面資料

3. 在 1.bmp 影像上畫兩條對角線(將線上的 pixel 值設為 255)。

Source Code：

```
#include <iostream>
#include "stdlib.h"
#include "bmp.h"
void diagonal(unsigned char **ima, unsigned char **bima, int nr,int nc);

using namespace std;

int main(int argc, char** argv) {
        unsigned char **ima, **bima;
        int nr,nc; //image height and width
        char filename[128],temp;
        bool isfilefine = false;
        //read bmp image from file
        cout << "Enter input filename:";
```

```
	cin >> filename;
	isfilefine = Read_BMP(filename, ima, nr, nc);
	if (!isfilefine)  return 0;
	bima=UC2D(nr,nc);
	Write_BMP_8bits("ima.bmp", ima, nr, nc);
	diagonal(ima,bima,nr,nc);
	Write_BMP_8bits("ex3_diagonal.bmp", bima, nr, nc);
	cout << "\nProgram done.\n";
	system("PAUSE");
	return 1;
}

void diagonal(unsigned char **ima, unsigned char **bima, int nr,int nc)
{
	for(int i=0;i<nr;i++)
		for(int j=0;j<nc;j++)
		{
			if(nr*j==nc*i || nr*(nc-1-j)==nc*i)
				bima[i][j]=255;
			else bima[i][j]=ima[i][j];
		}
}
```



(a) (b)

(a)原始影像 f1.bmp、(b) 畫兩條對角線影像 1.bmp

4. 取出 1.bmp 左上角 8x8 pixels 的數值資料存檔，再用 Excel 呈現資料表。

Source Code：

```
#include <iostream>
#include "stdlib.h"
#include "bmp.h"
#include<fstream>
void upper_left_excel(unsigned char **ima, unsigned char **bima);

using namespace std;

int main(int argc, char** argv) {
	unsigned char **ima, **bima;
	int nr,nc; //image height and width
	char filename[128],temp;
	bool isfilefine = false;

	//read bmp image from file
	cout << "Enter input filename:";
	cin >> filename;
	isfilefine = Read_BMP(filename, ima, nr, nc);
	if (!isfilefine)  return 0;
	bima=UC2D(nr,nc);
	Write_BMP_8bits("ima.bmp", ima, nr, nc);
	upper_left_excel(ima,bima);
	cout << "\nProgram done.\n";
	system("PAUSE");
	return 1;
}

void upper_left_excel(unsigned char **ima, unsigned char **bima)
{
	ofstream out("ex4_8x8pixels.xls");
	for(int i=0;i<8;i++){
		for(int j=0;j<8;j++)
		{
			out<<(int)ima[i][j]<<"\t";
		}
		out<<endl;
	}
}
```

1.bmp 左上角 8x8 pixels 的數值資料 :

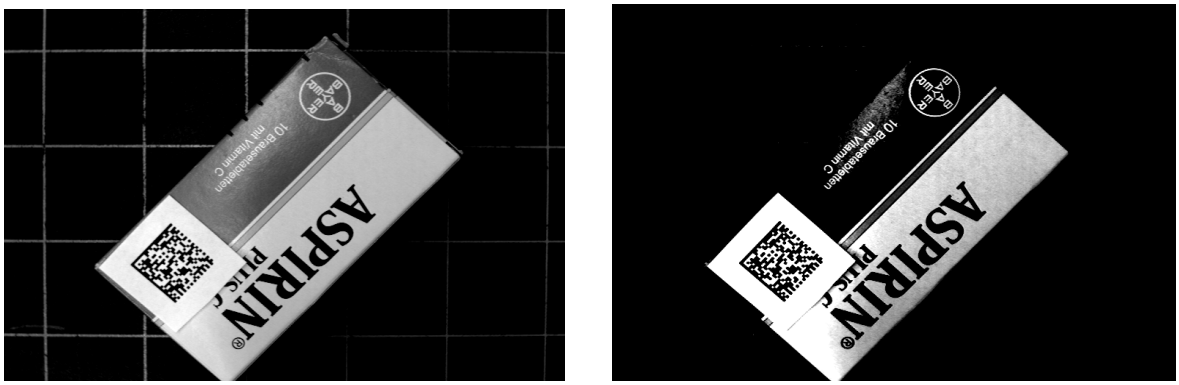
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

5. 改變 1.bmp 影像灰階分布，將小於 140 的灰階值全設為 0，將大於 203 的灰階值全設為 255，最後將 140-203 灰階值分布 平均擴展至 0-255。

Source Code：

```
#include <iostream>
#include "stdlib.h"
#include "bmp.h"

void gray_streching(unsigned char **ima, unsigned char **bima,int nr,int nc);

int main(int argc, char** argv) {

	unsigned char **ima, **bima;
	int nr,nc; //image height and width
	char filename[128],temp;
	bool isfilefine = false;

	//read bmp image from file
	cout << "Enter input filename:";
	cin >> filename;
	isfilefine = Read_BMP(filename, ima, nr, nc);
	if (!isfilefine)  return 0;
	bima=UC2D(nr,nc);
	Write_BMP_8bits("ima.bmp", ima, nr, nc);
	gray_streching(ima,bima,nr,nc);
	Write_BMP_8bits("ex5_gray_streching.bmp", bima, nr, nc);
	cout << "\nProgram done.\n";
	system("PAUSE");
	return 1;
}
```

```
void gray_streching(unsigned char **ima, unsigned char **bima,int nr,int nc)
{
	for(int i=0;i<nr;i++){
		for(int j=0;j<nc;j++)
		{
			if(ima[i][j]<140) bima[i][j]=0;
			else if(ima[i][j]>203) bima[i][j]=255;
			else bima[i][j]=(ima[i][j]-140)*255/63;
		}
	}
}
```



(a) (b)

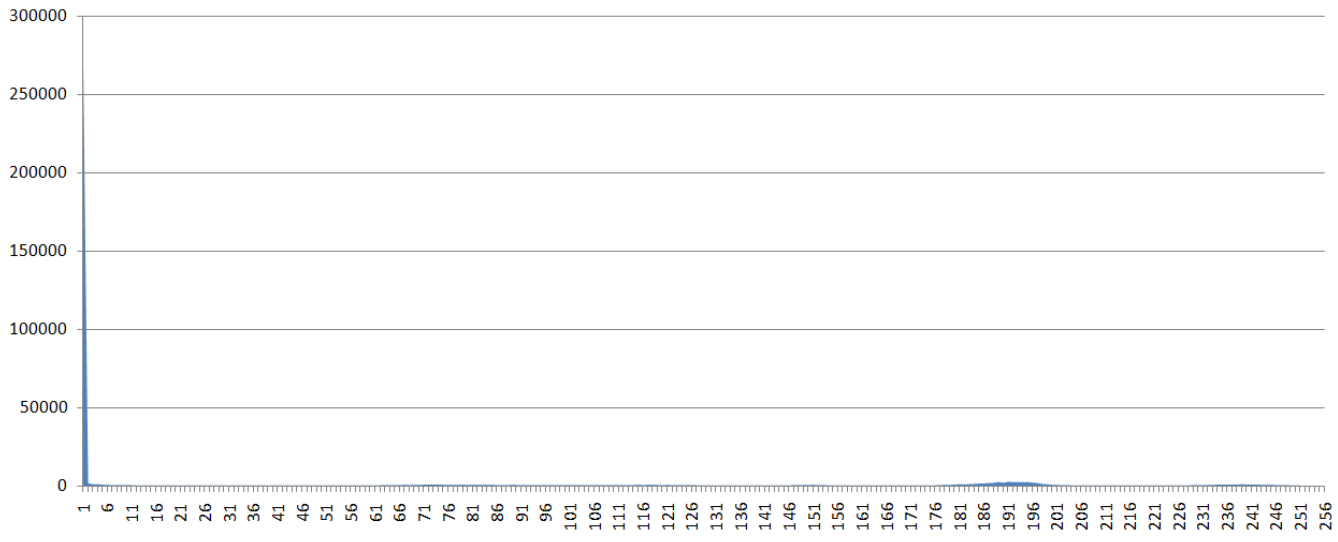
(a)原始影像 1.bmp、(b)處理後影像 ex5_gray_streching.bmp

6. 請找出一個最佳的 threshold 對 1.bmp 遙測影像作二值化。

Source Code：

```
#include <iostream>
#include "stdlib.h"
#include "bmp.h"
#include<fstream>
void binarize(unsigned char **ima, unsigned char **bima, int nr,int nc);

using namespace std;

int main(int argc, char** argv) {
	unsigned char **ima, **bima;
	int nr,nc; //image height and width
	char filename[128],temp;
	bool isfilefine = false;

	//read bmp image from file
	cout << "Enter input filename:";
```

```
	cin >> filename;
	isfilefine = Read_BMP(filename, ima, nr, nc);
	if (!isfilefine)  return 0;
	bima=UC2D(nr,nc);
	Write_BMP_8bits("ima.bmp", ima, nr, nc);
	binarize(ima,bima,nr,nc);
	Write_BMP_8bits("ex6_binarize.bmp", bima, nr, nc);
	cout << "\nProgram done.\n";
	system("PAUSE");
	return 1;
}

void binarize(unsigned char **ima, unsigned char **bima, int nr,int nc)
{
	ofstream out("ex6_histogram.txt");
	int histo[256]={0};
	for(int i=0;i<nr;i++)
		for(int j=0;j<nc;j++)
			histo[ima[i][j]]++;
	for(int i=0;i<256;i++)
		out<<i<<"\t"<<histo[i]<<endl;
	for(int i=0;i<nr;i++)for(int j=0;j<nc;j++)
	{
		if(ima[i][j]<150)bima[i][j]=0;
		else bima[i][j]=255;
	}
}
```



1.bmp 灰階分布圖

二值化影像 ex6_binarize.bmp

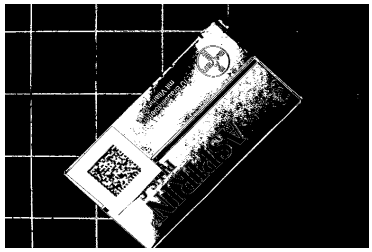
7. 輸出指紋影像 finger300x300 的 8 個 bit-planes。

```
#include <iostream>
#include "stdlib.h"
#include "bmp.h"
#include <string>
void bit_planes(unsigned char **ima, unsigned char **bima, int nr,int nc,int shift);

using namespace std;

int main(int argc, char** argv) {
	unsigned char **ima, **bima;
	int nr,nc; //image height and width
	char filename[128],temp;
	bool isfilefine = false;

	//read bmp image from file
	cout << "Enter input filename:";
	cin >> filename;
	isfilefine = Read_BMP(filename, ima, nr, nc);
	if (!isfilefine)  return 0;
	bima=UC2D(nr,nc);
	Write_BMP_8bits("ima.bmp", ima, nr, nc);
	string name[8]={
	"ex7_bit-plane0.bmp","ex7_bit-plane1.bmp","ex7_bit-plane2.bmp","ex7_bit-plane3.bmp",
	"ex7_bit-plane4.bmp","ex7_bit-plane5.bmp","ex7_bit-plane6.bmp","ex7_bit-plane7.bmp"
	};
	for(int shift=0;shift<8;shift++){
		bit_planes(ima,bima,nr,nc,shift);
		Write_BMP_8bits(name[shift].c_str(), bima, nr, nc);
```

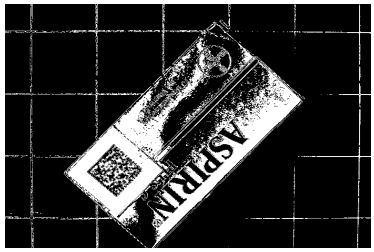
```
	}
	cout << "\nProgram done.\n";
	system("PAUSE");
	return 1;
}

void bit_planes(unsigned char **ima, unsigned char **bima, int nr,int nc,int shift)
{
	for(int i=0;i<nr;i++)
		for(int j=0;j<nc;j++)
			bima[i][j]=((ima[i][j]>>shift)%2)*255;
}
```

(a)原始影像 1.bmp

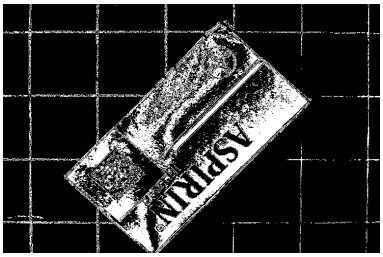
(b) ex7_bit-plane7.bmp



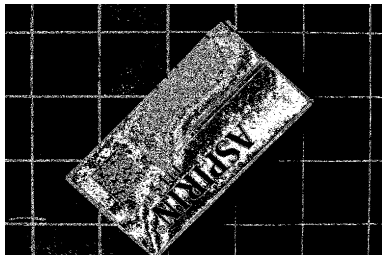
(c) ex7_bit-plane6.bmp



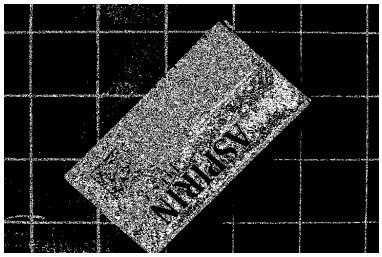
(d) ex7_bit-plane5.bmp



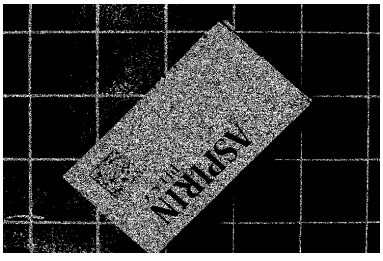
(e) ex7_bit-plane4.bmp



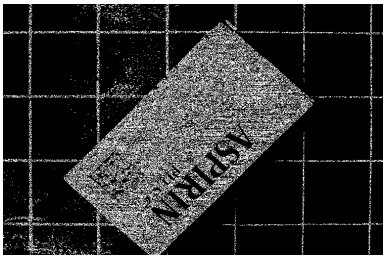
(f) ex7_bit-plane3.bmp



(g) ex7_bit-plane2.bmp



(h) ex7_bit-plane1.bmp



(i) ex7_bit-plane0.bmp

8. 將 1.bmp 影像 down-sampling 1/4 倍輸出。

Source Code：

```
#include <iostream>
#include "stdlib.h"
#include "bmp.h"
void down_sampling(unsigned char **ima, unsigned char **bima, int nr,int nc);

using namespace std;

int main(int argc, char** argv) {
	unsigned char **ima, **bima;
	int nr,nc; //image height and width
	char filename[128],temp;
	bool isfilefine = false;

	//read bmp image from file
	cout << "Enter input filename:";
	cin >> filename;
	isfilefine = Read_BMP(filename, ima, nr, nc);
	if (!isfilefine)  return 0;
	bima=UC2D(nr/4,nc/4);
	Write_BMP_8bits("ima.bmp", ima, nr, nc);

	down_sampling(ima,bima,nr/4,nc/4);
	Write_BMP_8bits("ex8_downsampling.bmp", bima, nr/4, nc/4);
	cout << "\nProgram done.\n";
	system("PAUSE");
	return 1;
}

void down_sampling(unsigned char **ima, unsigned char **bima, int nr,int nc)
{
	for(int i=0;i<nr;i++)
		for(int j=0;j<nc;j++)
			bima[i][j]=ima[4*i][4*j];
}
```

(a)

(b)

(a)原始影像 1.bmp、(b)down-sampling 1/4 影像 ex8_downsampling.bmp