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];  } |

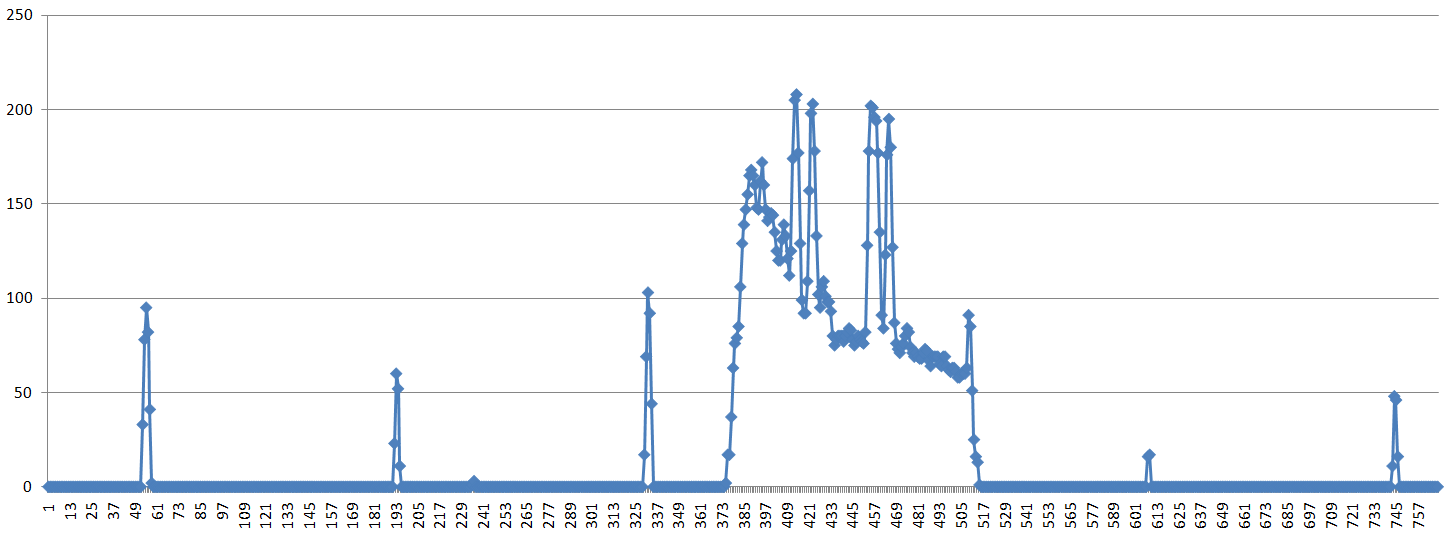
(a) (b) (c)

(a)原始圖檔1.bmp、(b)水平鏡射ex1\_horizontal.bmp、(c)垂直鏡射ex1\_vertical.bmp

1. 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

1. 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

1. 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 |

1. 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;  }  }  } |

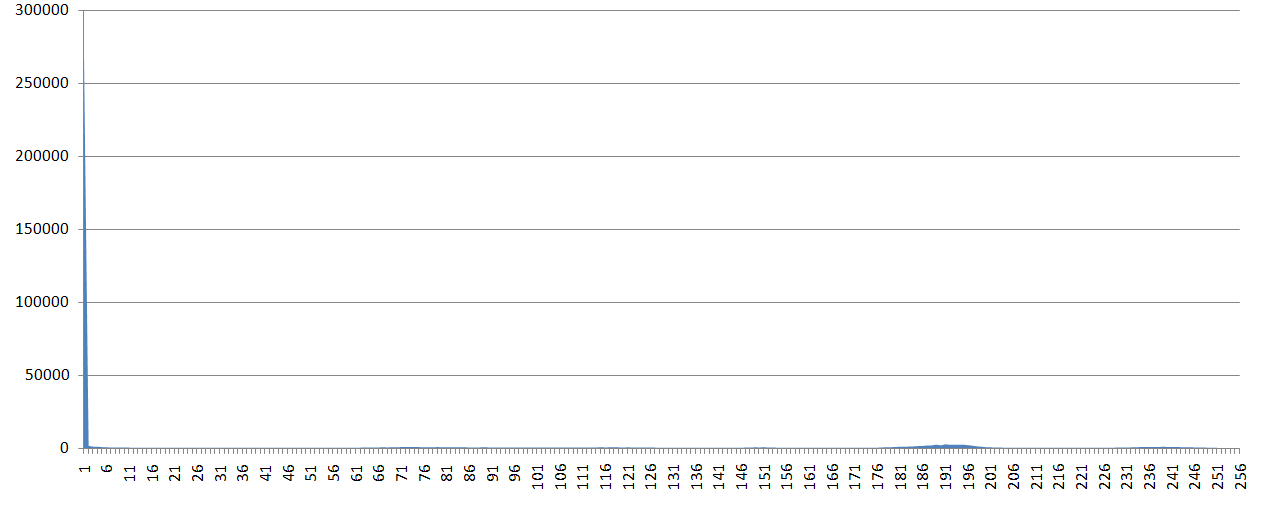
(a) (b)

(a)原始影像1.bmp、(b)處理後影像ex5\_gray\_streching.bmp

1. 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

1. 輸出指紋影像finger300x300的8個bit-planes。

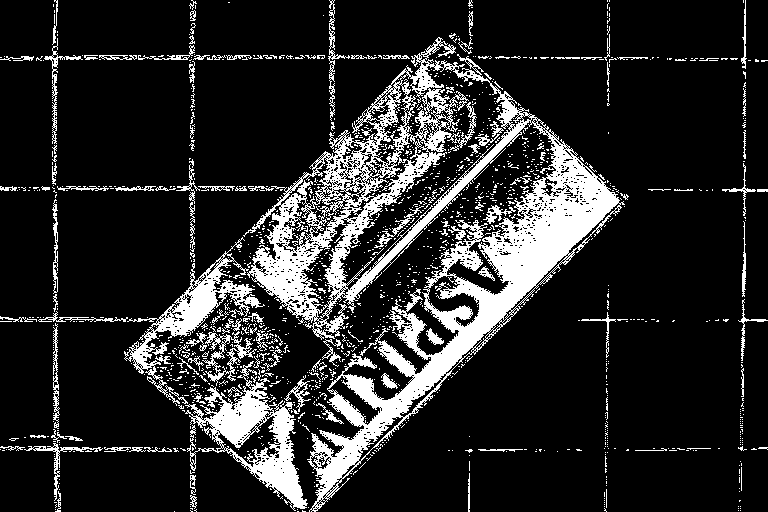
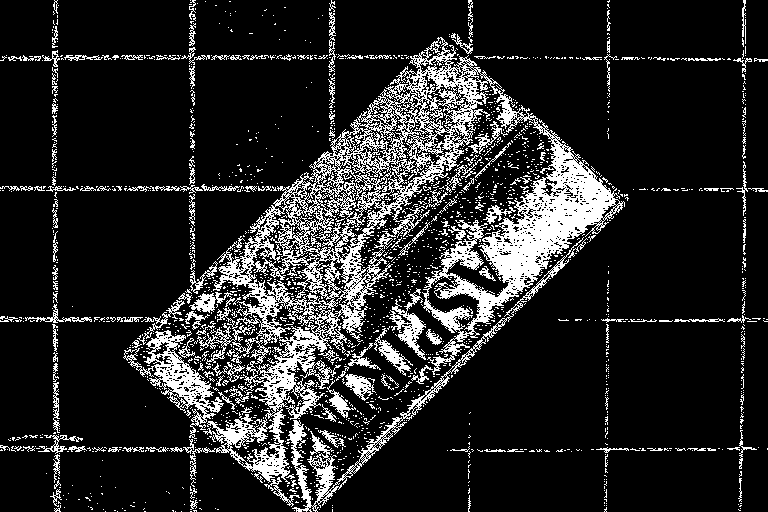
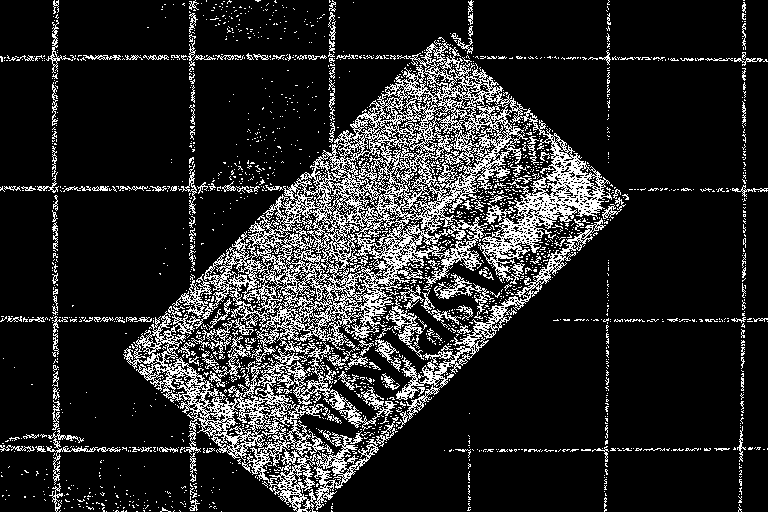
|  |
| --- |
| #include <iostream>  #include "stdlib.h"  #include "bmp.h"  #include <string>  void bit\_planes(unsigned char \*\*ima, unsigned char \*\*bima, int nr,int nc,int shift);  using namespace std;  int main(int argc, char\*\* argv) {  unsigned char \*\*ima, \*\*bima;  int nr,nc; //image height 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;  } |



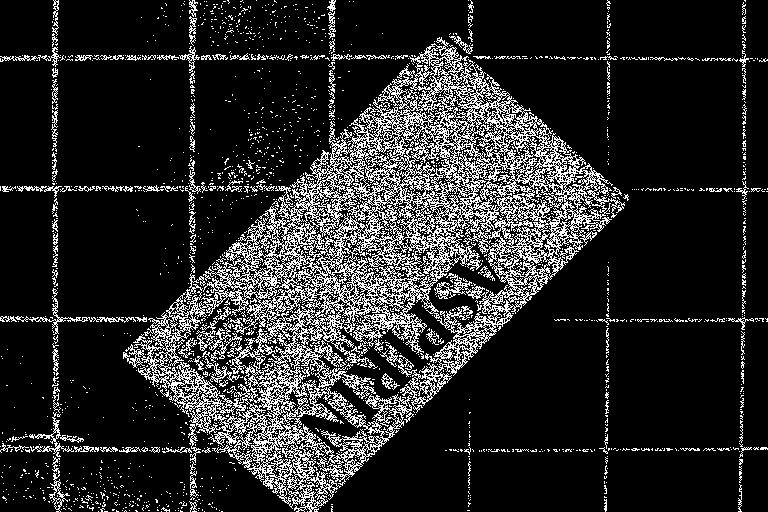
(a)原始影像1.bmp

(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

1. 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