

## 使用说明书

### 目录

PPM、PGM 和 PBM 图像处理.....	1
RAW 图像处理 .....	7
BMP 图像处理.....	9
其它处理.....	29

### PPM、PGM 和 PBM 图像处理

void OTSUBinarization(char* input, char* output)	OTSU 二值化。input 是输入文件名，output 是输出文件名。支持 P5 格式的 PGM 图像。
void SegmentsOTSUBinarization(char* input, char* output)	OTSU 二值化划分。input 是输入文件名，output 是输出文件名。支持 P5 格式的 PGM 图像。
void P3PPMBlur(char* input, char* output)	PPM 图像模糊，input 是输入文件名，output 是输出文件名。支持 P3 格式的 PPM 图像。
unsigned char** ReadPBM(char* input)	读取 PBM 图像并返回图像数据。input 是要读取的 PBM 图像文件名。支持 P4 格式的 PBM 图像。
void WritePBM(unsigned char** Input, char* output)	保存 PBM 图像。Input 是输入的图像数据，output 是输出文件名。支持 P4 格式的 PBM 图像。
void PGMSobel(char* input, char* output, int Mx[3][3], int My[3][3], int max, int min)	Sobel 算子，input 是输入文件名，output 是输出文件名。支持 P5 格式的 PGM 文件。 参考模板： int Mx[3][3] = {{-1, 0, 1}, {-2, 0, 2}, {-1, 0, 1}} int My[3][3] = {{-1, -2, -1}, {0, 0, 0}, {1, 2, 1}} int max = -9999 int min = 9999
void PGMSobelX(char* input, char* output, int Mx[3][3], int My[3][3], int max, int min)	X 方向滤波，input 是输入文件名，output 是输出文件名。支持 P5 格式的 PGM 文件。 参考模板： int Mx[3][3] = {{-1, 0, 1}, {-2, 0, 2}, {-1, 0, 1}} int My[3][3] = {{-1, -2, -1}, {0,

	0, 0}, {1, 2, 1}} int max = -9999 int min = 9999
void PGMSobelY(char* input, char* output, int Mx[3][3], int My[3][3], int max, int min)	Y 方向滤波, input 是输入文件名, output 是输出文件名。支持 P5 格式的 PGM 文件。 参考模板: int Mx[3][3] = {{-1, 0, 1}, {-2, 0, 2}, {-1, 0, 1}} int My[3][3] = {{-1, -2, -1}, {0, 0, 0}, {1, 2, 1}} int max = -9999 int min = 9999
void PGMSobel1(char* input, char* output, int min, int max, int mx[3][3], int my[3][3])	Sobel 算子, input 是输入文件名, output 是输出文件名。min 和 max 是 图像归一化相关参数, 如 min = 1000000, max = 0; mx 和 my 分别是 Sobel 算子的 X 和 Y 方向模板。支持 P2 和 P5 格式的 PGM。 参考模板 : int mx[3][3] = { {-1, 0, 1}, {-2, 0, 2}, {-1, 0, 1} }; int my[3][3] = { {-1, -2, -1}, {0, 0, 0}, {1, 2, 1} };
void PGMSobelX1(char* input, char* output, int min, int max, int mx[3][3], int my[3][3])	X 方向梯度, input 是输入文件名, output 是输出文件名。min 和 max 是 图像归一化相关参数, 如 min = 1000000, max = 0; mx 和 my 分别是 Sobel 算子的 X 和 Y 方向模板。支持 P2 和 P5 格式的 PGM。 参考模板 : int mx[3][3] = { {-1, 0, 1}, {-2, 0, 2}, {-1, 0, 1} }; int my[3][3] = { {-1, -2, -1},

	<pre> {0, 0, 0}, {1, 2, 1} }; </pre>
<pre> void PGMSobelY1(char* input, char* output, int min, int max, int mx[3][3], int my[3][3]) </pre>	<p>Y 方向梯度，input 是输入文件名，output 是输出文件名。min 和 max 是图像归一化相关参数，如 min = 1000000, max = 0; mx 和 my 分别是 Sobel 算子的 X 和 Y 方向模板。支持 P2 和 P5 格式的 PGM。</p> <p>参考模板：</p> <pre> int mx[3][3] = {     {-1, 0, 1},     {-2, 0, 2},     {-1, 0, 1} }; int my[3][3] = {     {-1, -2, -1},     {0, 0, 0},     {1, 2, 1} }; </pre>
<pre> void PGMSobel2(char* input, char* XOutput, char* YOutput, char* SobelOutput, int sobel_x[3][3], int sobel_y[3][3], int min, int max) </pre>	<p>Sobel 算子，input 是输入文件名，output 是输出文件名。支持 P5 格式的 PGM 图像。XOutput 是输出的 X 方向的梯度图像，YOutput 是输出的 Y 方向的梯度图像，SobelOutput 是输出的整幅图像的 Sobel 算子计算结果，min 和 max 是图像归一化的相关参数，如 min=100, max=0。</p> <p>参考模板：</p> <pre> int sobel_x[3][3]={{-1, 0, 1}, {-2, 0, 2}, {-1, 0, 1}}; int sobel_y[3][3]={{1, 2, 1}, {0, 0, 0}, {-1, -2, -1}}; </pre>
<pre> void HorizSobel(char* input, char* output) </pre>	<p>水平 Sobel 算子，input 是输入文件名，output 是输出文件名。支持 P5 格式的 PGM 图像。</p>
<pre> void VertSobel(char* input, char* output) </pre>	<p>垂直 Sobel 算子，input 是输入文件名，output 是输出文件名。支持 P5 格式的 PGM 图像。</p>
<pre> void PGMSobel1(char* input, char* output, int threshold) </pre>	<p>Sobel 算子，input 是输入文件名，output 是输出文件名。支持 P5 格式的 PGM 图像。threshold 是目标阈值，如 threshold=80。</p>
<pre> void </pre>	<p>直方图均衡化，input 是输入文件名，</p>

PGMHistogramEqualization(char* input, char* output)	output 是输出文件名。支持 P5 格式的 PGM 图像。
PPMImage* ReadPPM(char* input)	PPM 图像读取, input 是要读取的 PPM 图像文件名。支持 P6 格式的 PPM 图像。 需要引入的结构体: typedef struct { unsigned char red, green, blue;     //像素的颜色由 RGB (红/绿/蓝) 表示 } PPMPixel; typedef struct { unsigned int width, height; // 图像的宽度和高度 (以像素为单位) PPMPixel *data; // 构成图像的像素 } PPMImage;
void WritePPM(char* output, PPMImage* img)	PPM 图像保存, output 是输出的 PPM 图像文件名, img 是输入的图像数据。支持 P6 格式的 PPM 图像。 需要引入的结构体: typedef struct { unsigned char red, green, blue;     //像素的颜色由 RGB (红/绿/蓝) 表示 } PPMPixel; typedef struct { unsigned int width, height; // 图像的宽度和高度 (以像素为单位) PPMPixel *data; // 构成图像的像素 } PPMImage;
void InvertColor(char* input, char* output)	负滤波器, input 是输入文件名, output 是输出文件名。支持 P6 格式的 PPM 图像。
void GrayFilter(char* input, char* output)	灰度过滤器, input 是输入文件名, output 是输出文件名。支持 P6 格式的 PPM 图像。
void SepiaFilter(char* input, char* output)	乌贼墨过滤器, input 是输入文件名, output 是输出文件名。支持 P6 格式的 PPM 图像。
void AdjustSaturation(char* input, char* output, double a)	调整图像饱和度, input 是输入文件名, output 是输出文件名。a 是目标饱

	和度, 如 a=30。支持 P6 格式的 PPM 图像。
void Resize(char* input, char* output, unsigned int NewWidth, unsigned int NewHeight)	调整图像大小, input 是输入文件名, output 是输出文件名。NewWidth 和 NewHeight 分别是输出图像的宽和高。支持 P6 格式的 PPM 图像。
void AdjustHue(char* input, char* output, int a)	调整图像的色调, input 是输入文件名, output 是输出文件名。a 是目标色调, 如 a=125。支持 P6 格式的 PPM 图像。
void AdjustBrightness(char* input, char* output, double a)	调整图像亮度, input 是输入文件名, output 是输出文件名。a 是目标亮度, 如 a=60。支持 P6 格式的 PPM 图像。
void AdjustContrast(char* input, char* output, double a)	调整图像对比度, input 是输入文件名, output 是输出文件名。a 是目标对比度, 如 a=60。支持 P6 格式的 PPM 图像。
void AdjustBlur(char* input, char* output, double a)	通过 sigma 因子模糊图像, input 是输入文件名, output 是输出文件名。a 是 sigma 因子, 如 a=5。支持 P6 格式的 PPM 图像。
void MeanGrayFilter(char* input, char* output, double a)	平均灰度滤波器, input 是输入文件名, output 是输出文件名。a 是平均值系数, 如 a=3。支持 P6 格式的 PPM 图像。
void Pixelate(char* input, char* output, unsigned int a)	像素化, input 是输入文件名, output 是输出文件名。a 是幅度值, 如 a=8。支持 P6 格式的 PPM 图像。
void Rotate(char* input, char* output, short a)	旋转图像, input 是输入文件名, output 是输出文件名。a 是旋转的角度, 如 a=45。支持 P6 格式的 PPM 图像。
void GammaCorrection(char* input, char* output, double a)	伽马校正, input 是输入文件名, output 是输出文件名。a 是 gamma 数, 如 a=0.5。支持 P6 格式的 PPM 图像。
void GrayAndChannelSeparation(char* input, char* Grayoutput, char* Routput, char* Goutput, char* Boutput)	生成灰度图以及 RGB 通道分离, input 是输入的 P6 格式的 PPM 图像; Grayoutput 是输出的灰度图文件名, Routput、Goutput 和 Boutput 分别是输出的 R、G 和 B 通道的图像文件名, 输出都是 PGM 格式文件。
void PGMBin(char* input, char* output, int threshold)	灰度图像二值化, 输入是灰度图像, 输入和输出都是 PGM 文件, threshold 是阈值, 如 threshold=125。

void Brightening(char* input, char* output, int a)	彩色图像增亮, 输入和输出都是 P6 格式的 PPM 图像, a 是增亮系数, 如 a=80。
void GrayBrightening(char* input, char* output, int a)	灰度图像增亮, 输入和输出都是 PGM 图像, a 是增亮系数, 如 a=80。
void PPMFilter(char* input, char* output)	彩色图像滤波, 输入和输出都是 P6 格式的 PPM 文件。
void PGMGrayFilter(char* input, char* output)	灰度图像滤波, 输入和输出都是 PGM 图像。
void PPMtoBMP(char* input, char* output)	PPM 图像转 BMP 图像, input 是输入文件名, output 是输出文件名。支持 P6 格式的 PPM 图像。
void YFiltering(char* input, char* output, int sobel_x[3][3], int sobel_y[3][3])	Y 方向滤波, input 是输入文件名, output 是输出文件名。支持 P5 格式的 PGM 图像。 参考模板: <pre>int sobel_x[3][3] = { { 1, 0, -1},                       { 2, 0, -2},                       { 1, 0, -1}}; int sobel_y[3][3] = { { 1, 2, 1},                       { 0, 0, 0},                       {-1, -2, -1}};</pre>
void XFiltering(char* input, char* output, int sobel_x[3][3], int sobel_y[3][3])	X 方向滤波, input 是输入文件名, output 是输出文件名。支持 P5 格式的 PGM 图像。 参考模板: <pre>int sobel_x[3][3] = { { 1, 0, -1},                       { 2, 0, -2},                       { 1, 0, -1}}; int sobel_y[3][3] = { { 1, 2, 1},                       { 0, 0, 0},                       {-1, -2, -1}};</pre>
void SobelFiltering(char* input, char* output)	Sobel 算子, input 是输入文件名, output 是输出文件名。支持 P5 格式的 PGM 图像。

input, char*                      output, int sobel_x[3][3], int sobel_y[3][3])	output 是输出文件名。支持 P5 格式的 PGM 图像。 参考模板： int sobel_x[3][3] = { { 1,    0, -1},  {    2, 0,   -2},  {    1, 0,   -1}}; int sobel_y[3][3] = { { 1, 2,    1},  {    0, 0,    0},  {-1, -2,   -1}};
void            PrewittFiltering(char* input, char*                      output, int prewitt_x[3][3], int prewitt_y[3][3])	Prewitt 算子, input 是输入文件名, output 是输出文件名。支持 P5 格式的 PGM 图像。 参考模板： int prewitt_x[3][3] = { { 5, 5, 5},  {    -3, 0, -3},  { -3, - 3, -3}}; int prewitt_y[3][3] = { { 5, -3, -3},  { 5, 0, -3},  {5, -3, -3}};
void        LaplacianFiltering(char* input, char*                      output, int laplacian[3][3])	Laplace 算子, input 是输入文件名, output 是输出文件名。支持 P5 格式的 PGM 图像。laplacian 是 Laplacian 算子模板。 参考模板： int laplacian[3][3] = { { 1,    1, 1},  { 1,   - 8,   1},  {    1, 1,   1}};
void            PGMotsuThreshold(string input, char* output)	大津阈值法, input 是输入文件名, output 是输出文件名。支持 P5 格式的

	PGM 图像。
void PGMLocalisedOtsuThreshold(string input, char* output)	局部大津阈值，input 是输入文件名， output 是输出文件名。支持 P5 格式的 PGM 图像。
void PGMSauvolaThreshold(string input, char* output, double a, double b, double c)	索沃拉阈值，input 是输入文件名， output 是输出文件名。a、b 和 c 的参 考值如：a=0.01, b=15, c=225。支持 P5 格式的 PGM 图像。
void PGMThreshold(string input, char* output, int thresh)	阈值法，input 是输入文件名，output 是输出文件名。支持 P5 格式的 PGM 图 像。thresh 是阈值，如：thresh=5。

## RAW 图像处理

void RAWSobelEdge(char* input, char* output, int ROWS, int COLS, int M, float sobelX[3][3], float sobelY[3][3])	Sobel 算子，input 是输入文件名， output 是输出文件名。ROWS 是图像 的行，COLS 是图像的列，M 是滤波相 关参数，如 M=1。支持 RAW 图像。 参考模板： float sobelX[3][3] = {{-1, 0, 1}, {- 2, 0, 2}, {- 1, 0, 1}};  float sobelY[3][3] = {{-1, - 2, -1},  {0, 0, 0},  {1, 2, 1}};
void RAWPlaceholder(char* input, char* output, int ROWS, int COLS, int M, float mask[3][3])	边缘检测，input 是输入文件名， output 是输出文件名。ROWS 是图像 的行，COLS 是图像的列，M 是滤波相 关参数，如 M=1。支持 RAW 图像。 参考模板： float mask[3][3] = {{-1, -2, -1},  {0, 0, 0},  {1, 2, 1}};
void RAWLaplacialSharpeningFilter(char * input, char* output, int ROWS, int COLS, int M, float w, float	拉普拉斯锐化滤波器，input 是输入 文件名，output 是输出文件名。ROWS 是图像的行大小，COLS 是图像的列大 小，M 和 w 是滤波相关参数，如 M=1，



mask[3][3])	<p>w=1; mask 是滤波器模板。支持 RAW 图像。</p> <p>参考模板：</p> <pre>float mask[3][3] = {{0, 1, 0},                     {1, -4, 1},                     {0, 1, 0}};</pre>
void RawLaplacianEnhancement(char* input1, char* output1, int width, int height)	<p>拉普拉斯算子增强, input1 是输入的 RAW 图像文件名, output1 是输出的 RAW 图像文件名, width 是输入图像的宽, height 是输入图像的高。支持 RAW 图像。</p>
void RawPowerTransformation(char* input, char* output, int width, int height, int c, float v)	<p>幂次变换, input 是输入的 RAW 图像文件名, output 是输出的 RAW 图像文件名, width 是输入图像的宽, height 是输入图像的高。默认 c=1, v=0.6。支持 RAW 图像。</p>
void RAWAvgFilter(char* input, char* output, int ROWS, int COLS, int M, float mask[3][3])	<p>平均滤波器, input 是输入文件名, output 是输出文件名。ROWS 是图像的行大小, COLS 是图像的列大小, M 是滤波相关参数, 如 M=1; mask 是滤波器模板。支持 RAW 图像。</p> <p>参考模板：</p> <pre>float mask[3][3] = {{0.1111, 0.1111, 0.1111},                     {0.1111, 0.1111, 0.1111},                     {0.1111, 0.1111, 0.1111}};</pre>
void RawImageInversion(char* input, char* output, int width, int height)	<p>图像反相, input 是输入的 RAW 图像文件名, output 是输出的 RAW 图像文件名, width 是输入图像的宽, height 是输入图像的高。支持 RAW 图像。</p>
void RawHistogramEqualization(char* input, char* output, int width, int height)	<p>直方图均衡化, input 是输入的 RAW 图像文件名, output 是输出的 RAW 图像文件名, width 是输入图像的宽, height 是输入图像的高。支持 RAW 图像。</p>
void RAWMedianFilter(char* input, char* output, int ROWS, int COLS, int M, int sequence[9])	<p>中值滤波, input 是输入文件名, output 是输出文件名。ROWS 是图像的行, COLS 是图像的列, M 是滤波相关参数, 如 M=1。支持 RAW 图像。</p> <p>参考模板：</p> <pre>int sequence[9] = {0, 0, 0, 0, 0, 0, 0, 0, 0}</pre>



void           EdgeDetection4(char* input, char* output, int a)	边缘检测，input 是输入文件名，output 是输出文件名。a 是用于设置图像像素的相关参数，如 a=3。支持 24 位 BMP 图像。
void    GrayImageConversion8(char* input, char* output)	生成图像的灰度图，支持 8 位 BMP 图像。input 是输入文件名，output 是输出文件名。
void    Gray(char*       input, char* output)	灰度图转换，支持 24 位 BMP 图像。input 是输入文件名，output 是输出文件名。
void    GrayImageConversion(char* input, char* output)	彩色图转灰度图，input 是要处理的彩色图像，output 是处理后生成的灰度图名称。支持 24 位 BMP 图像。
void    OTSU(char*       input, char* output, int BeforeThreshold)	大津算法，input 是输入文件名，output 是输出文件名。BeforeThreshold 是初始阈值，如 BeforeThreshold=10。支持 8 位 BMP 图像。
void           LowerBrightness(char* input, char* output, int a, int b)	调低亮度，input 是输入文件名，output 是输出文件名。支持 24 位 BMP 图像。a 和 b 的参考值可为 a=100, b=0。
void           HightBrightness(char* input, char* output, int a, int b)	调高亮度，input 是输入文件名，output 是输出文件名。支持 24 位 BMP 图像。a 和 b 的参考值可为 a=100, b=0。
void    IterativeThresholdSelection(char* input, char* output)	迭代阈值选择，input 是输入文件名，output 是输出文件名。支持 8 位 BMP 图像。
void           DitheringMethod(char* input, char* output)	抖动法，input 是输入文件名，output 是输出文件名。支持 8 位 BMP 图像。
void           LogTransformation(char* input, char* output, int constant)	对数变换，input 是输入文件名，output 是输出文件名。支持 8 位 BMP 图像。constant 是相关参数，如 constant=15。
void    LogarithmicTransformation(char* input, char* output)	对数变换，input 是输入文件名，output 是输出文件名。支持 BMP 图像。
void    HistogramEqualization(char* input, char* output)	直方图均衡化，input 是输入文件名，output 是输出文件名。支持 BMP 图像。
void           Binarization(char* input, char* output, int threshold)	二值化，input 是输入文件名，output 是输出文件名。支持 24 位 BMP 图像。threshold 是 阈 值 ， 如 :

	threshold=128。
void LowPassFilter(char* input, char* output)	低通滤波器, input 是输入文件名, output 是输出文件名。支持 BMP 图像。
void HighPassFilter(char* input, char* output)	高通滤波器, input 是输入文件名, output 是输出文件名。支持 BMP 图像。
void Thinning(char* input, char* output)	图像细化, input 是输入文件名, output 是输出文件名。支持 BMP 图像。
void ThinningLine(char* input, char* output)	图像细化且线条化, input 是输入文件名, output 是输出文件名。支持 BMP 图像。
void Corrosion(char* input, char* output)	腐蚀, input 是输入文件名, output 是输出文件名。支持 4 位 BMP 图像。
void Corrosion1(char* input, char* output, int *TempBuf, int TempH, int TempW)	腐蚀, input 是输入文件名, output 是输出文件名。支持 24 位 BMP 图像。TempBuf 是腐蚀模板, TempH 和 TempW 分别是 TempBuf 的高和宽, 如 TempH=4, TempW=4, 则有 TempBuf[4][4]。
void Expand(char* input, char* output, int *TempBuf, int TempH, int TempW)	膨胀, input 是输入文件名, output 是输出文件名。支持 24 位 BMP 图像。TempBuf 是膨胀模板, TempH 和 TempW 分别是 TempBuf 的高和宽, 如 TempH=4, TempW=4, 则有 TempBuf[4][4]。
void GaussianFiltering(char* input, char* output)	高斯滤波, input 是输入文件名, output 是输出文件名。支持 24 位 BMP 图像。
void LaplaceEnhancement(char* input, char* output)	拉普拉斯增强, input 是输入文件名, output 是输出文件名。支持 24 位 BMP 图像。
void Residual(char* input, char* output)	求残差, input 是输入文件名, output 是输出文件名。支持 24 位 BMP 图像。
void Compress(char* input, char* output)	压缩, 支持多种文件。input 是要压缩的文件名, output 是压缩后的文件名。
void Decompression(char* input, char* output)	解压缩, 支持多种文件。input 是要解压缩的文件名, output 是解压缩后的文件名。
void BlackWhite(char* input, char* output)	黑白化, input 是输入的原图像, output 是输出的黑白图像。支持 24 位 BMP 图像。

void Underexposure(char* input, char* output)	图像欠曝光, input 是输入的原图像, output 是输出的欠曝光图像。支持 24 位 BMP 图像。
void Overexposure(char* input, char* output)	图像过曝光, input 是输入的原图像, output 是输出的过曝光图像。支持 24 位 BMP 图像。
void GammaTransform(char* input, char* output)	伽马变换, input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。
void GrayScale(char* input, char* output)	灰度化, input 是输入文件名, output 是输出文件名。支持 24 位 BMP 图像。
void GrayImageBinarization(char* input, char* output, int bit, int threshold)	灰度图二值化, bit 用于设定位数, 如 bit=8; threshold 是阈值, 如 threshold=200。支持 8 位 BMP 图像。
void GreyPseudoColor(char* input, char* output)	灰度图伪彩色化, input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。
void CalculateCumulativeHistogramMap(char* input, char* outfile)	计算累加直方图并映射, input 是输入文件名, outfile 是输出文件名。支持 24 位 BMP 图像。
void Translation(string input, char* output, int dx, int dy)	图像平移, input 是输入的文件, dx 和 dy 是横向及纵向的移动距离 (像素), 负值是向左 / 向下移动; output 是平移操作后的结果文件名。支持 BMP 图像。
void Mirrored(string input, char* output, char axis)	镜像变换, input 是输入的文件, output 是镜像操作后的结果文件名, axis 是镜像变换的方向 (以 X 或 Y 表示)。支持 BMP 图像。
void Sheared(string input, char* output, char axis, double Coef)	错切变换, input 是输入的文件, output 是错切操作后的结果文件名, axis 和 Coef 分别是错切变换的方向 (以 X 或 Y 表示) 和错切系数, 负值是向左 / 向下偏移。支持 BMP 图像。
void Scaled(string input, char* output, double cx, double cy)	缩放操作, input 是输入的文件, output 是缩放操作后的结果文件名, cx 和 cy 分别是横向及纵向的缩放系数, 系数大于 1 表示拉伸, 小于 1 表示压缩。支持 BMP 图像。
void Rotated1(string input, char* output, double angle)	图像旋转, input 是输入的文件, output 是图像旋转后的结果文件名, angle 是旋转角度, 弧度制。支持 BMP 图像。
void SaltNoise(char* input, char*	添加椒盐噪声, a 和 b 是噪声相关参



	<pre> 2},                                 {1,    2,                                 1}}; </pre>
<pre> void    SharpeningFiltering(char* input, char*          output, short average[3][3], short sharpen[3][3]) </pre>	<p>锐化滤波，input 是输入文件名，output 是输出文件名。支持 8 位 BMP 图像。</p> <p>参考模板：</p> <pre> short average[3][3] = {{1, 2, 1},                         {2,    4,                         2},                         {1,    2,                         1}}; short sharpen[3][3] = {{-1, -1,                         -1},                         {-1, 8, -                         1},                         {-1, -1,                         -1}}; </pre>
<pre> void    GradientSharpening(char* input, char*          output, short average[3][3], short soble1[3][3], short soble2[3][3]) </pre>	<p>梯度锐化，input 是输入文件名，output 是输出文件名。支持 8 位 BMP 图像。</p> <p>参考模板：</p> <pre> short average[3][3] = {{1, 2, 1},                         {2,    4,                         2},                         {1,    2,                         1}}; short soble1[3][3] = {{-1, -2, -                         1},                         {0, 0, 0},                         {1,    2,                         1}}; short soble2[3][3] = {{-1, 0, 1},                         {-2,    0,                         2},                         {-1,    0,                         1}}; </pre>
<pre> void    ArithmeticMeanFilter(char* input, char* output) </pre>	<p>算术平均滤波器，input 是输入文件名，output 是输出文件名。支持 8 位 BMP 图像。</p>
<pre> void    GeometricMeanFilter(char* input, char* output) </pre>	<p>几何平均滤波器，input 是输入文件名，output 是输出文件名。支持 8 位 BMP 图像。</p>
<pre> void    HarmonicMeanFilter(char* </pre>	<p>调和平均滤波器，input 是输入文件</p>

input, char* output)	名, output 是输出文件名。支持 8 位 BMP 图像。
void ContraHarmonicMeanFilter(char* input, char* output)	反调和平均滤波器, input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。
void Filter(char* input, char* output)	滤波, input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。
void Mosaic(char* input, char* output, int x)	马赛克化图像, input 是输入文件名, output 是输出文件名。x 是马赛克处理的块的大小。支持 24 位 BMP 图像
void Expansion(char* input, char* output)	膨胀, input 是输入文件名, output 是输出文件名。支持 4 位 BMP 图像。
void SmoothSharpen(char* input, char* output, int Template[3][3], int coefficient)	平滑, input 是输入文件名, output 是输出文件名。Template 是平滑模板, 均一化处理, coefficient1 = 9。支持 24 位 BMP 图像。
void GaussSmoothSharpen(char* input, char* output, int Template[3][3], int coefficient)	高斯平滑, input 是输入文件名, output 是输出文件名。Template 是高斯平滑模板, coefficient=16。支持 24 位 BMP 图像。
void SobelSharpen(char* input, char* output, int Templatex[3][3], int Templatey[3][3], int coefficient1, int coefficient2)	Sobel 算子, input 是输入文件名, output 是输出文件名。Templatex 是 laplace 锐化模板, 4 邻域, Templatey 是 laplace 锐化模板, 8 邻域, coefficient1 = 9, coefficient2 = 16。支持 24 位 BMP 图像。
void MidSmoothing(char* input, char* output)	中值滤波器, input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像
void AvgSmoothing(char* input, char* output)	均值滤波器, input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。
void Averaging(char* input1, char* input2, char* input3, char* output, int a)	图像平均化, input 是输入文件名, output 是输出文件名。a 是平均化相关参数, 如 a=3。支持 8 位 BMP 图像。
void PlaneSlicing(char* input, char* output)	平面切片, input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。
void SharpeningSpatialFiltering8(char* input, char* output, int model[9])	锐化空间滤波器, input 是输入文件名, output 是输出文件名。model 是锐化模板。支持 8 位灰度图像。
void PseudoGrayscale(char* input, char* output)	伪灰度化, input 是输入文件名, output 是输出文件名。支持 24 位 BMP 图像。



void TwoColors(char* input, char* output, int threshold, unsigned char color1, unsigned char color2)	二色化, input 是输入文件名, output 是输出文件名。threshold 是阈值, 如 threshold=115; color1 和 color2 是要填充的两个颜色。支持 24 位 BMP 图像。
void PNGImageGeneration(char* filename, const unsigned char img[], unsigned W, unsigned H, int x)	filename 是生成的 PNG 图像文件名; img 是图像的像素数据, W 是图像的宽, H 是图像的高, x=0 选择生成 RGB 图像, x=1 选择生成 RGBA 图像。
void BilateralFiltering(string input, char* output, double ssd, double sdd)	双边滤波, input 是输入文件名, output 是输出文件名。支持 24 位 BMP 图像。ssd 和 sdd 分别是空间域标准差与强度域标准差。
int Equal(char* input1, char* input2, double c)	若比对图像的梯度幅相似性偏差值等于 c 则通过。input1 和 input2 是要比对的两个图像。c 是参考的阈值。支持 24 位 BMP 图像。
int GreaterThan(char* input1, char* input2, double c)	若比对图像的梯度幅相似性偏差值大于 c 则通过。input1 和 input2 是要比对的两个图像。c 是参考的阈值。支持 24 位 BMP 图像。
int LessThan(char* input1, char* input2, double c)	若比对图像的梯度幅相似性偏差值小于 c 则通过。input1 和 input2 是要比对的两个图像。c 是参考的阈值。支持 24 位 BMP 图像。
double GMSD(char* input1, char* input2)	求两幅图像的梯度幅相似性偏差值并返回结果。input1 和 input2 是要比对的两个图像。支持 24 位 BMP 图像。
void AddGaussNoise(char* input, char* output)	添加高斯噪声, input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。
void AddSaltPepperNoise(char* input, char* output)	添加椒盐噪声, input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。
void ChannelSeparation(char* input, char* Routput, char* Goutput, char* Boutput)	通道分离, input 是输入文件名, Routput 是红色通道图像, Goutput 是绿色通道图像, Boutput 是绿色通道图像。支持 24 位 BMP 图像。
void PatternMethod(char* input, char* output, unsigned char Template[8][8])	图案法, input 是输入文件名, output 是输出文件名。Template 是模板数组。支持 8 位 BMP 图像。
void BMP24LossyCompression(char* input, char* output)	图像有损压缩, input 是待压缩的 BMP 文件名, output 是有损压缩后输出的

	文件名。支持 24 位 BMP 图像。
void BMP24LossyDecompression(char* input, char* output)	图像有损解压, input 是待解压的文件名, output 是输出解压后的 BMP 文件名。支持 24 位 BMP 图像。
void BMP24LosslessCompression(char* input, char* output)	图像无损压缩, input 是待压缩的 BMP 文件名, output 是无损压缩后输出的文件名。支持 24 位 BMP 图像。
void BMP24LosslessDecompression(char* input, char* output)	图像无损解压, input 是待解压的文件名, output 是输出解压后的 BMP 文件名。支持 24 位 BMP 图像。
void       ImageDiscoloration(char* input, char* output, double a, double b, double c)	图像变色, input 是输入文件名, output 是输出文件名。如: a=0.2126, b=0.7152, c=0.0722。支持 24 位 BMP 图像。
void               ImageCutting(char* input, char*               output, int leftdownx, int           leftdowny, int rightupx, int rightupy)	图像裁剪, input 是输入文件名, output 是输出文件名。leftdownx, leftdowny, rightupx, rightupy 是要裁剪的矩形区域的左下角和右上角的坐标(连续四个整数值, 如 50 50 300 300)。支持 24 位 BMP 图像。
void       RGBtoGraywithoutLUT(char* input, char* output)	图像无 LUT 的灰度化, input 是输入文件名, output 是输出文件名。支持 24 位 BMP 图像。
void       RGBtoGraywithLUT(char* input, char* output)	图像有 LUT 的灰度化, input 是输入文件名, output 是输出文件名。支持 24 位 BMP 图像。
void PiecewiseLinearTransform(char* input, char* output)	分段线性变换, input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。
void       PowerConversion(char* input, char* output, double c, double g)	功率转换, input 是输入文件名, output 是输出文件名。如: c = 1.2, g = 0.5。支持 8 位 BMP 图像。
void   LaplacianEnhancement(char* input, char*   output, int   N, int LaplMask[3][3])	拉普拉斯图像增强, input 是输入文件名, output 是输出文件名。如: N=1。支持 8 位 BMP 图像。 参考模板: int LaplMask[3][3] = { 0, 1, 0, 1, -4, 1, 0, 1, 0 };
void   Smooth(char*   input, char* output)	平滑, input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。
void               LaplaceSmooth(char*	拉普拉斯平滑, input 是输入文件名,

input, char*     output, int     N, int LaplMask[3][3])	output 是输出文件名。如：N=1。支持 8 位 BMP 图像。 参考模板： <pre>int LaplMask[3][3] = {     0, 1, 0,     1, -4, 1,     0, 1, 0 };</pre>
void     Sobel1(char*     input, char* output, int     N, int SblMask1[3][3], int SblMask2[3][3])	Sobel 算子，input 是输入文件名， output 是输出文件名。如：N=1。支持 8 位 BMP 图像。 参考模板： <pre>int SblMask1[3][3] = {     -1, -2, -1,     0, 0, 0,     1, 2, 1 }; int SblMask2[3][3] = {     -1, 0, 1,     -2, 0, 2,     -1, 0, 1 };</pre>
void SobelSmooth(char* input, char* output, int     N, int SblMask1[3][3], int SblMask2[3][3])	Sobel 平滑，input 是输入文件名， output 是输出文件名。如：N=1。支持 8 位 BMP 图像。 参考模板： <pre>int SblMask1[3][3] = {     -1, -2, -1,     0, 0, 0,     1, 2, 1 }; int SblMask2[3][3] = {     -1, 0, 1,     -2, 0, 2,     -1, 0, 1 };</pre>
void     Multiply(char*     input, char* output, int     N, int SblMask1[3][3], int SblMask2[3][3], int LaplMask[3][3])	图像倍增化，input 是输入文件名， output 是输出文件名。如：N=1。支持 8 位 BMP 图像。 参考模板： <pre>int LaplMask[3][3] = {     0, 1, 0,     1, -4, 1,     0, 1, 0 };</pre>

	<pre> }; int SblMask1[3][3] = {     -1, -2, -1,     0, 0, 0,     1, 2, 1 }; int SblMask2[3][3] = {     -1, 0, 1,     -2, 0, 2,     -1, 0, 1 }; </pre>
<pre> void Add(char* input, char* output, int N, int SblMask1[3][3], int SblMask2[3][3], int LaplMask[3][3]) </pre>	<p>图像添加，input 是输入文件名，output 是输出文件名。如：N=1。支持 8 位 BMP 图像。</p> <p>参考模板：</p> <pre> int LaplMask[3][3] = {     0, 1, 0,     1, -4, 1,     0, 1, 0 }; int SblMask1[3][3] = {     -1, -2, -1,     0, 0, 0,     1, 2, 1 }; int SblMask2[3][3] = {     -1, 0, 1,     -2, 0, 2,     -1, 0, 1 }; </pre>
<pre> void PowerConversion1(char* input, char* output, double c, double g, int N, int SblMask1[3][3], int SblMask2[3][3], int LaplMask[3][3]) </pre>	<p>功率变换，input 是输入文件名，output 是输出文件名。如：c = 1.2, g = 0.5, N=1。支持 8 位 BMP 图像。</p> <pre> int LaplMask[3][3] = {     0, 1, 0,     1, -4, 1,     0, 1, 0 }; int SblMask1[3][3] = {     -1, -2, -1,     0, 0, 0,     1, 2, 1 }; int SblMask2[3][3] = { </pre>

	-1, 0, 1, -2, 0, 2, -1, 0, 1 };
void BlackWhite(char* input, char* output)	黑白化图像, input 是输入文件名, output 是输出文件名。支持 24 位 BMP 图像。
void RandomOperation(char* input, char* output, unsigned char threshold1, unsigned char threshold2, unsigned char threshold3, unsigned char threshold4, unsigned char threshold5, unsigned char threshold6, unsigned char red, unsigned char green, unsigned char blue, int color1, int color2, int color3, int color4, int color5, int color6, int color7, int color8)	随意操作, input 是输入文件名, output 是输出文件名。支持 24 位 BMP 图像。
void SpecialEffects1(char* input, char* output, unsigned char red, unsigned char green, unsigned char blue)	图像特效, input 是输入文件名, output 是输出文件名。支持 24 位 BMP 图像。
void GrayAVS(char* input, char* output, float k, float b)	input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。
void HistogramEqualize24(char* input, char* output)	直方图均衡化, input 是输入文件名, output 是输出文件名。支持 24 位 BMP 图像。
void AvrFilter(char* input, char* output1, char* output2, int M, int N)	input 是输入文件名, output 是输出文件名。如 M=21, N=1。支持 8 位 BMP 图像。
void GryOppositionSSE(char* input, char* output)	input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。
void MedianFilter(char* input, char* output, int M, int N)	中值滤波器, input 是输入文件名, output 是输出文件名。如 M=5, N=5。支持 8 位 BMP 图像。
void EdgeSharpeningGry(char* input, char* output)	input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。
void SJGryandRiceTest(char* input, char* output)	input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。
void TextTest(char* input, char* output)	input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。
void RedChannel(char* input, char*	生成图像的红色通道图像, input 是

output)	输入文件名，output 是输出文件名。支持 24 位 BMP 图像。
void GreenChannel(char* input, char* output)	生成图像的绿色通道图像，input 是输入文件名，output 是输出文件名。支持 24 位 BMP 图像。
void BlueChannel(char* input, char* output)	生成图像的蓝色通道图像，input 是输入文件名，output 是输出文件名。支持 24 位 BMP 图像。
void HistogramStatistics(char* input, char* output)	直方图统计，input 是输入文件名，output 是输出文件名。支持 24 位 BMP 图像。
void HistogramEqualization1(char* input, char* output)	直方图均衡化，input 是输入文件名，output 是输出文件名。支持 24 位 BMP 图像。
void ReflectionRay(char* input, char* output)	反射线，input 是输入文件名，output 是输出文件名。支持 24 位 BMP 图像。
void MeanFiltering24(char* input, char* output)	均值滤波，input 是输入文件名，output 是输出文件名。支持 24 位 BMP 图像。
void MedianFiltering24(char* input, char* output)	中值滤波，input 是输入文件名，output 是输出文件名。支持 24 位 BMP 图像。
void ZoomOutAndZoomIn(char* input, char* output, double value)	缩放（双线性插值），input 是输入文件名，output 是输出文件名。value 是放大倍数，如 value=0.5。支持 24 位 BMP 图像。
void Translation24(char* input, char* output, int x, int y)	平移，input 是输入文件名，output 是输出文件名。x 是横轴的平移量，y 是纵轴的平移量，如 x=-10, y=-30。支持 24 位 BMP 图像。
void Mirror24(char* input, char* output)	镜像，input 是输入文件名，output 是输出文件名。支持 24 位 BMP 图像。
void Rotate24(char* input, char* output, double degree)	旋转，input 是输入文件名，output 是输出文件名。degree 是旋转的度数。支持 24 位 BMP 图像。
void GivenThresholdMethod(char* input, char* output, int threshold)	给定阈值法处理图像，使图片黑白化，input 是输入文件名，output 是输出文件名。threshold 是给定的阈值，如 threshold=100。支持 24 位 BMP 图像。
void IterativeThresholdMethod(char* input, char* output)	迭代阈值法处理图像，使图片黑白化，input 是输入文件名，output 是输出文件名。支持 24 位 BMP 图像。
void	Ostu（大津法）阈值分割，input 是输

OstuThresholdSegmentationMethod(char* input, char* output)	入文件名, output 是输出文件名。支持 24 位 BMP 图像。
void Repudiation(char* input, char* output)	将伪彩图片反白, input 是输入文件名, output 是输出文件名。支持 24 位 BMP 图像。
void Gray1(char* input, char* output)	将彩色图片变成灰度图片, input 是输入文件名, output 是输出文件名。支持 24 位 BMP 图像。
void CorrectMethod(char* input, char* output)	正确法, input 是输入文件名, output 是输出文件名。支持 24 位 BMP 图像。
void ChannelSeparation1(char* input, char* Routput, char* Goutput, char* Boutput)	对图像分理出其中的 RGB 分量并分别保存为独立的图像, input 是输入文件名, Routput 是红色通道图像, Goutput 是绿色通道图像, Boutput 是绿色通道图像。支持 24 位 BMP 图像。
void ReverseColor(char* input, char* output)	对灰度图进行反色, input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。
Image* LoadImage1(char* input)	BMP 图像读取, input 是输入文件名。支持 8 位和 24 位 BMP 图像。 返回 Image 型数据, Image 型数据的结构如下: typedef struct { int width; int height; int channels; //图像通道数 unsigned char* Data; //像素数据 }Image;
void SaveImage1(char* output, Image* img)	将 Image 型数据保存为 BMP 图像, output 是生成的 BMP 图像文件名, img 是要保存的图像数据。支持 8 位和 24 位 BMP 图像。 Image 型数据的结构如下: typedef struct { int width; int height; int channels; //图像通道数 unsigned char* Data; //像素数据 }Image;
void ImageContrastExtension(char*	图像对比度扩展, input 是输入文件

input, char* output, double m, double g1, double g2, double a)	<p>名, output 是输出文件名。</p> <p>其中, 可参考: double m=1.5, g1=100.0, g2=200.0; m 对应斜率</p> $\text{double } a = (255.0 - m * (g2 - g1)) / (255.0 - (g2 - g1));$ <p>支持 8 位 BMP 图像。</p>
void Binaryzation(char* input, char* output, int threshold)	图像二值化, input 是输入文件名, output 是输出文件名。threshold 是将灰度值转化为二值的阈值, 如 threshold=80。支持 24 位 BMP 图像。
void GlobalBinarization(char* input, char* output)	全局二值化, input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。
void AdaptiveBinarization(char* input, char* output)	自适应二值化, input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。
void ExpansionOperation(char* input, char* output)	膨胀操作, input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。
void CorrosionOperation(char* input, char* output)	腐蚀操作, input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。
void Operation1(char* input, char* output)	开操作, input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。
void Closed1(char* input, char* output)	闭操作, input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。
void Negative1(char* input, char* output)	图像反色, input 是输入文件名, output 是输出文件名。支持 24 位 BMP 图像。
void Negative(char* input, char* output)	图像反色, input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。
void ImageEncryption(char* inFileName, char* outFileName, char key)	图像加密, inFileName 是原图图像文件名, outFileName 是解密图像文件名, key 是密钥, 如 key=255。支持 8 位、24 位和 32 位 BMP 图像。
void ImageDecryption(char* inFileName, char* outFileName, char key)	图像解密, inFileName 是加密图像文件名, outFileName 是解密图像文件名, key 是密钥, 如 key=255。支持 8 位、24 位和 32 位 BMP 图像。
void EncryptionDecryption(char* input, char* output, int Key, int a)	图像加解密, Key 是密钥, a=1 时执行加密, a=0 时执行解密。支持 24 位 BMP 图像。



void Encryption(char* input, char* output, int Key)	图像加密，input 是输入文件名，output 是输出文件名。Key 是密钥。支持 24 位 BMP 图像。
void Decryption(char* input, char* output, int Key)	图像解密，input 是输入文件名，output 是输出文件名。Key 是密钥。支持 24 位 BMP 图像。
void Compress8(string input, string output)	图像压缩，input 是输入文件名，output 是输出文件名。支持 8 位 BMP 图像。
void Decompression(string input, string output)	图像解压，input 是输入文件名，output 是输出文件名。支持 8 位 BMP 图像压缩后的结果文件。
void HorizontalMirror(char* input, char* output)	水平镜像，input 是输入文件名，output 是输出文件名。支持 8 位 BMP 图像。
void MirrorVertically(char* input, char* output)	垂直镜像，input 是输入文件名，output 是输出文件名。支持 8 位 BMP 图像。
void XMirroring(char* input, char* output)	X 镜像，input 是输入文件名，output 是输出文件名。支持 8 位 BMP 图像。
void YMirroring(char* input, char* output)	Y 镜像，input 是输入文件名，output 是输出文件名。支持 8 位 BMP 图像。
void ImageConvolution(char* input, char* output, double** Kernel, int n, int m)	图像卷积，input 是输入文件名，output 是输出文件名。Kernel 是卷积核，如 double Kernel[3][3] = {{-0.225, -0.225, -0.225}, {-0.225, 1, -0.225}, {-0.225, -0.225, -0.225}}; n 是 Kernel 的第一维的大小，m 是 Kernel 的第二维的大小，形如 Kernel[n][m]。支持 24 位 BMP 图像。
void Translation(char* input, char* output, int x, int y, unsigned char color)	图像平移，input 是输入文件名，output 是输出文件名。x 和 y 是在 X 轴和 Y 轴平移的量，以右为正向，color 是平移后非原图区域填充的颜色，如 color=100。支持 8 位 BMP 图像。
void Nesting(char* Biginput, char* Smallinput, char* output)	图像嵌套，Biginput 是输入的大图，Smallinput 是输入的小图。支持 24 位 BMP 图像。
void Blend(char* input1, char* input2, char* output)	图像融合之混合化，input1 和 input2 是输入的两个要融合的图像，output 是输出文件名。支持 24 位 BMP 图像。
void Checker(char* input1, char* input2, char* output)	图像融合之棋盘化，input1 和 input2 是输入的两个要融合的图像，output

	是输出文件名。支持 24 位 BMP 图像。
void Blend1(char* input1, char* input2, char* output)	图像融合之混合化, input1 和 input2 是输入的两个要融合的图像, output 是输出文件名。支持 24 位 BMP 图像。
void Checker1(char* input1, char* input2, char* output)	图像融合之棋盘化, input1 和 input2 是输入的两个要融合的图像, output 是输出文件名。支持 24 位 BMP 图像。
void ImageSharpening(char* input, char* output)	图像锐化, input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。
void DrawRectangle(char* input, char* output, int x1, int y1, int x2, int y2, unsigned char red, unsigned char green, unsigned char blue)	在 24 位 BMP 图像上通过传入的参数画一个矩形。input 是输入文件名, output 是输出文件名。(x1, y1) 是矩形坐上顶点的坐标, (x2, y2) 是矩形右下顶点的坐标; red 是矩形线框的红色分量, green 是矩形线框的绿色分量, blue 是矩形的蓝色分量。
void GenerateBmp(unsigned char* pData, int width, int height, char* filename)	生成 BMP 图像, pData 是图像的像素数据, width 和 height 是图像的宽和高, filename 是生成的图像的文件名。
void Jpg24ImageGeneration(char* filename, unsigned int width, unsigned int height, unsigned char* img)	JPG 图像生成, filename 是生成的 JPG 图像文件名, width 是图像的宽, height 是图像的高, img 是图像的像素数据。
void ImageScalingNearestNeighborInterpolation(char* input, char* output, float lx, float ly)	最近邻插值法去栅格, input 是输入文件名, output 是输出文件名。lx 和 ly 是长和宽需要缩放的倍数。支持 8 位 BMP 图像。
void ImageScalingBilinearInterpolation(char* input, char* output, float lx, float ly)	双线性插值法去栅格, input 是输入文件名, output 是输出文件名。lx 和 ly 是长和宽需要缩放的倍数。支持 8 位 BMP 图像。
void BilinearInterpolationScaling(char* input, char* output, float ExpScalValue)	双线性插值, input 是输入文件名, output 是输出文件名。ExpScalValue 是期望的缩放倍数(允许小数)。支持 BMP 图像。
void NearestNeighborInterpolationScaling(char* input, char* output, float ExpScalValue)	最近邻插值, input 是输入文件名, output 是输出文件名。ExpScalValue 是期望的缩放倍数(允许小数)。支持 BMP 图像。
void RotateRight90Degrees(char* input, char* output)	input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像, 向右旋转 90 度。

void RotateLeft90Degrees(char* input, char* output)	input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像, 向左旋转 90 度。
void ImageRotation(char* input, char* output, double angle)	图像旋转, input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。angle 是要旋转的角度。
void Rotation8(char* input, char* output, double Angle, int x1, int y1, int x2, int y2, unsigned char color)	图像旋转, input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。Angle 是要旋转的角度数; x1、y1、x2、y2 是旋转所围绕的中心点的坐标, color 是旋转后非原图区域的填充颜色。
void Rotation24(char* input, char* output, double Angle, int x1, int y1, int x2, int y2, unsigned char red, unsigned char green, unsigned char blue)	图像旋转, input 是输入文件名, output 是输出文件名。支持 24 位 BMP 图像。Angle 是要旋转的角度数; x1、y1、x2、y2 是旋转所围绕的中心点的坐标; red、green、blue 分别是旋转后非原图区域要填充的颜色的红绿蓝分量。
void Rotation(char* input, char* output, int angle, unsigned char color)	图像旋转, input 是输入文件名, output 是输出文件名。支持 8 位 BMP 图像。angle 是旋转的角度, color 是旋转后非原图区域填充的颜色, 如 color=100。
void Rotate(char* input, char* output, int angle)	图像旋转, input 是输入文件名, output 是输出文件名。支持 BMP 图像。angle 是旋转的角度。
void FileWrite(char* BMP, char* TXT)	图像隐写之文件写入, 将文本文件写入图像。支持 32 位 BMP 图像。BMP 是要写入的图像文件名, TXT 是要写入图像的文本文件名。
void FileWriteOut(char* BMP, char* TXT)	图像隐写之文件写出, 将文本文件从图像中取出来。支持 32 位 BMP 图像。BMP 是要写出的图像文件名, TXT 是写出图像后信息保存的文本文件名。
void grayToColor(FILE* input, FILE* output)	灰色转伪彩色, input 是输入文件, output 是输出文件。支持 8 位和 24 位 BMP 图像。
void ImageThinning(char* input, char* output, char** str, int n, int m1, int a, int b)	图像细化, input 是输入文件名, output 是输出文件名。支持 4 位 BMP 图像。n 是 str 第一维的大小, m1 是第二维的大小, 形如 str[n][m1]; a 和 b 是相关的调节参数, 可以为 a=3, b=5。

	<p>参考模板：</p> <pre>char str[6][8] = { { 0, 0, 0, 0, 0, 0, 0, 0 }, { 255, 0, 255, 0, 0, 255, 0, 0 },     { 255, 0, 255, 255, 0, 255, 0, 255 }, { 255, 255, 255, 0, 0, 255, 255, 255 },     { 255, 0, 255, 255, 0, 255, 255, 255 }, { 0, 255, 255, 255, 255, 255, 255, 255 } };</pre>
int MinimumValueOfImagePixels(char* filename)	返回图像像素的最小值，filename 是输入的图像文件名。支持 8 位和 24 位 BMP 图像。
int MaximumValueOfImagePixels(char* filename)	返回图像像素的最大值，filename 是输入的图像文件名。支持 8 位和 24 位 BMP 图像。
float AverageValueOfImagePixels(char* filename)	返回图像像素的均值，filename 是输入的图像文件名。支持 8 位和 24 位 BMP 图像。
double StandardDeviationOfImagePixels(char* filename)	返回图像像素的标准差，filename 是输入的图像文件名。支持 8 位和 24 位 BMP 图像。
double EntropyOfImage(char* filename)	返回图像的熵，支持 8 位和 24 位 BMP 图像。
float* CountTheFrequencyOfPixels(char* filename)	filename 是输入的图像文件名。存储每个像素的频率，像素值为 0~255，返回值数组中的元素序号即为像素值，该序号在数组下的值即为这个像素的频率。支持 8 位和 24 位 BMP 图像。
void HistogramEqualization2(char* input, char* output, int imgBit)	直方图均衡化，input 是输入文件名，output 是输出文件名。支持 8 位和 24 位 BMP 图像。imgBit 是输入图像的位数。
void HistogramEqualization3(char* input, char* output)	直方图均衡化，input 是输入文件名，output 是输出文件名。支持 8 位和 24 位 BMP 图像。
void HistogramEqualization4(char* input, char* output)	直方图均衡化，input 是输入文件名，output 是输出文件名。支持 8 位和 24 位 BMP 图像。input 是输入文件名称，out 是输出文件名称。
void MedianFiltering1(char* input, char* output)	中值滤波，input 是输入文件名，output 是输出文件名。支持 8 位 BMP 图像。

void MedianFiltering2(char* input, char* output)	中值滤波，input 是输入文件名，output 是输出文件名。支持 8 位和 24 位 BMP 图像。
void ThresholdProcessing(char* input, char* output, int Threshold)	阈值处理，input 是输入文件名，output 是输出文件名。支持 8 位 BMP 图像。Threshold 是阈值相关参数，如 Threshold=0.001
void OTSUProcessing(char* input, char* output)	大津法处理，input 是输入文件名，output 是输出文件名。支持 8 位 BMP 图像。
void HistogramEqualization5(char* input, char* output)	直方图均衡，支持 8 位和 16 位 BMP。
void Resize(char* input, char* output, int Height, int Width)	图片缩放，支持 8 位和 16 位 BMP。Height 和 Width 是输出图像的高和宽。
double MeanBrightness(char* input)	求图像的平均亮度，支持 8 位和 16 位 BMP。
void KMeans(string input, unsigned int Clusters, char* output)	K-Means 聚类，input 是输入文件名，Clusters 是聚类的种类数目，output 是输出文件名。支持 PNG 等多种图像。

## 其他处理

void DES_Encrypt(char *PlainFile, char *Key, char *CipherFile)	DES 加密函数，支持多种文件。PlainFile 是原文件的文件名，Key 是密钥字符，CipherFile 是加密后的文件名。
void DES_Decrypt(char *CipherFile, char *Key, char *PlainFile)	DES 解密函数，支持多种文件。CipherFile 是已加密文件的文件名，Key 是密钥字符，PlainFile 是解密后的文件名。
void Encode(char* input, char* output)	文本文件压缩，input 是输入文件名，output 是输出文件名。
void Decode(char* input, char* output)	文本文件压缩结果解压缩，input 是输入文件名，output 是输出文件名。
void FileCompress(char *input , char *output)	文件压缩，input 是输入文件名，output 是输出文件名。
void FileDecompression(char *input , char *output)	文件压缩结果解压缩，input 是输入文件名，output 是输出文件名。