double CharacterRecognition(char* TargetImage,char* TemplateFileGroup[])	Character matching, supports BMP images, and the return value is the sequence number of the template file matched to the target image. If the return value is 2, it indicates that the image matches the template with sequence number 2 (starting from zero). reference : TemplateFileGroup[]={ "0.txt", "1.txt", "2.txt", "3.txt", "4.txt", "5.txt", "6.txt", "7.txt", "8.txt", "9.txt" };
double CharacterRecognition1(char* TargetImage,char* TemplateFileGroup[])	Character matching, supports BMP images, and the return value is the sequence number of the template file matched to the target image. If the return value is 2, it indicates that the image matches the template with sequence number 2 (starting from zero). reference : TemplateFileGroup[]={ "0.txt", "1.txt", "2.txt", "3.txt", "4.txt", "5.txt", "6.txt", "7.txt", "8.txt", "9.txt" };
void CodeEncoding(std::string input,char* output, int width,int height, int margin, int eccLevel, int stride_bytes, int comp,int a)	QR code and barcode encoding. input is the string to be encoded, and output is the file name of the generated QR code image. Margin: The margin around the barcode ECC: Error correction level, [0-8] a=1：AZTEC a=2：CODABAR a=3：CODE_39 a=4：CODE_93 a=5：CODE_128 a=6：DATA_MATRIX a=7：EAN_8 a=8：EAN_13 a=9：ITF a=10：MAXICODE

	a=11：PDF_417 a=12：QR_CODE a=13：RSS_14 a=14：RSS_EXPANDED a=15：UPC_A a=16：UPC_E a=17：UPC_EAN_EXTENSION Reference：margin=10，eccLevel=-1，stride_bytes=0，comp=1。
std::string CodeDecoding(char* input,int req_comp,int a)	Decoding QR codes and barcodes. input is the file name of the input QR code image, and returns the decoding result. a=1：Lum a=2：RGB a=3：BGR a=4：RGBX a=5：XRGB a=6：BGRX a=7：XBGR Reference：req_comp=4，a=4。