# 情报督察

## 源代码

### codebook\_generator.cpp

1. #include <fstream>
2. #include <iostream>
3. #include <string>
5. **int** main(**int** argc, **char** \*argv[])
6. {
7. **if** (argc != 2)
8. {
9. std::cout << "Usage: " << argv[0] << " codebook2.txt\n";
10. **return** 1;
11. }
12. srand(1);
13. **int** codebook[256] = {}, temp1, temp2;
14. **for** (**int** i = 0; i < 256; i++)
15. {
16. codebook[i] = i;
17. }
18. **for** (**int** i = 0; i < 256; i++)
19. {
20. temp1 = rand() % 256;
21. temp2 = rand() % 256;
22. codebook[temp1] ^= codebook[temp2] ^= codebook[temp1] ^= codebook[temp2];
23. }
25. std::string codebook\_name = argv[1];
26. std::ofstream outfile(codebook\_name, std::ios\_base::out | std::ios\_base::binary);
27. **for** (**int** i = 0; i < 256; i++)
28. {
29. std::cout << codebook[i] << ' ';
30. outfile << codebook[i] << ' ';
31. }
33. outfile.close();
35. **return** 0;
36. }

### encrypt.cpp

1. #include <iostream>
2. #include <string>
3. #include <fstream>
4. #include <sstream>
6. **int** main(**int** argc, **char** \*argv[])
7. {
8. **if**(argc != 4)
9. {
10. std::cout << "Usage: " << argv[0] << " codebook infile outfile\n";
11. **return** 1;
12. }
13. std::string codebookFile\_name = argv[1], inFile\_name = argv[2], outFile\_name = argv[3], msgs, encrypt\_msgs;
14. std::ifstream codebookFile(codebookFile\_name), inFile(inFile\_name);
15. std::ofstream outFile(outFile\_name);
16. std::ostringstream outbuff;
17. **int** codebook[256], i = 0;
18. **while**(!codebookFile.eof())
19. {
20. codebookFile >> msgs;
21. // std::cout << msgs << ' ';
22. codebook[i] = std::atoi(msgs.c\_str());
23. // std::cout << codebook[i] << ' ';
24. ++i;
25. }
27. **while**(!inFile.eof())
28. {
29. inFile >> msgs;
30. //     // std::cout << msgs << '\n';
32. **for** (auto it = msgs.begin(); it != msgs.end(); ++it)
33. {
34. **int** temp = **static\_cast**<**int**>(\*it);
35. outbuff << codebook[temp] << ' ';
36. }
37. outbuff << codebook[' '] << ' ';
38. }
39. encrypt\_msgs = outbuff.str();
40. std::cout << encrypt\_msgs << '\n';
41. outFile << encrypt\_msgs;
42. inFile.close();
43. outFile.close();
44. }

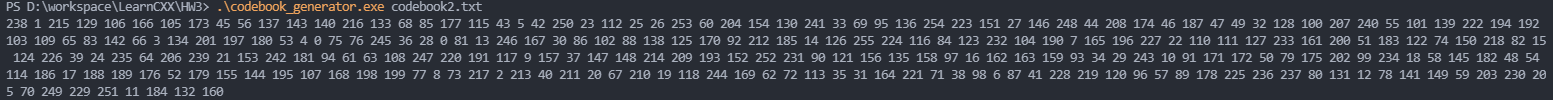
### decrypt.cpp

1. #include <iostream>
2. #include <string>
3. #include <fstream>
4. #include <sstream>
6. **int** main(**int** argc, **char** \*argv[])
7. {
8. **if**(argc != 4)
9. {
10. std::cout << "Usage: " << argv[0] << " codebook infile outfile\n";
11. **return** 1;
12. }
13. **int** decodebook[256] = {};
14. std::string codebookFile\_name = argv[1], inFile\_name = argv[2], outFile\_name = argv[3];
15. std::ifstream codebookFile(codebookFile\_name), inFile(inFile\_name);
16. std::ofstream outFile(outFile\_name);
17. std::ostringstream outbuff, decodebookbuff;
18. std::string msgs, decrypt\_msgs;
20. **for** (**int** i = 0; i < 256; ++i)
21. {
22. codebookFile >> msgs;
23. decodebook[std::atoi(msgs.c\_str())] = i;
24. // std::cout << decodebook[std::atoi(msgs.c\_str())] << '\n';
25. }

28. **while**(!inFile.eof())
29. {
30. inFile >> msgs;
31. // std::cout << msgs << '\n';
32. outbuff << **static\_cast**<**char**>(decodebook[std::atoi(msgs.c\_str())]);
33. }
34. decrypt\_msgs = outbuff.str();
35. std::cout << decrypt\_msgs << '\n';
36. outFile << decrypt\_msgs;
37. inFile.close();
38. outFile.close();
39. }

## 运行结果

### 文字加密

码本生成：

原文：hello c++ 123 456

密文：

翻译：

### 文件加密

Win10下不知道怎么生成二进制文件