



深蓝学院  
shenlanxueyuan.com

## Project III 情报督察作业分享

主讲人 蓝猫



- 第一部分：读取密码本
- 第二部分：编码
- 第三部分：译码

# 读取密码本

读取两类密码：

1. 字符串：例如qwertyuiopasdfghjklzxcvbnm

2. 数字：例如 43 65 131 188 236 230 143 169 20 177 151 4 217 90 237 18 100 22 126 200  
158 198 33 12 199 144 107 48 60 120 62 223 93 176 81 173 204 208 28 47 153 192 66 102  
89 190 155 112 219 68 201 134 124 82 16 111 244 26 127 6 163 106 160 187 97 194 80 59  
2 215 180 85 34 227 228 203 116 119 54 129 94 178 96 172 103 51 182 25 40 113 183 77  
174 83 247 57 61 142 50 11 139 254 243 226 13 191 87 69 147 55 150 121 241 235 233  
246 15 9 3 52 64 248 149 181 210 46 24 53 186 44 212 86 32 71 136 17 146 67 8 195 171 1  
185 49 118 91 98 56 92 104 41 242 114 206 224 138 7 175 75 234 0 105 159 167 193 128  
39 84 36 130 231 141 251 197 31 79 73 168 164 189 239 214 42 88 154 101 205 135 196  
137 209 202 95 221 250 148 14 45 10 35 166 213 108 78 23 238 162 19 110 207 252 179  
249 211 157 21 216 63 76 255 37 133 245 140 165 30 240 99 27 109 225 74 161 125 170  
38 115 156 218 72 58 123 232 253 122 220 229 145 152 5 70 117 222 184 29 132

# 字符串读取

```
std::string read_codebook_char(std::string& path){  
    std::ifstream in(path);  
    std::string code_string;  
    getline(in, code_string);  
    if (code_string.back() == '\\r'){  
        code_string.erase(code_string.size() - 1);  
    }  
    return code_string;  
}
```

'\\r': MacOS中的换行符

# 数字读取

```
std::vector<int> read_codebook_number(std::string& path){  
    std::ifstream file_handler(path);  
    std::vector<int> arr;  
    int number;  
    while (file_handler >> number) {  
        arr.push_back(number);  
        file_handler.ignore(std::numeric_limits<std::streamsize>::max(), ' ');  
    }  
    return arr;  
}
```

读取文件的时候，只读取数字，忽视空格。

# 纲要

---

- 第一部分：读取密码本
- **第二部分：编码**
- 第三部分：译码

```
code_string = read_codebook_char(path_codebook);
int n = code_string.length();
std::ifstream input(path_input);
std::ofstream output(path_output);
if (input.is_open()) {
    for (char c; input.get(c);){
        if ((c > 'a') && (c <= 'z')){
            c = code_string[c - 'a'];
        }
        output.put(c);
    }
}
else{
    std::cout << "Error Opening Input!" << std::endl;
}
```

注意:

1. 考虑文件打开出现异常的情况，使用 `is_open()`。
2. 使用 `codebook[c - 'a']` 找到对应的密码。

```
std::vector<int> array;  
array = read_codebook_number(path_codebook);  
  
unsigned char codebook [256];  
for (int i = 0; i < 256; ++i){  
    codebook[i] = static_cast<unsigned char>(array[i]);  
}  
  
std::ifstream input(path_input);  
std::ofstream output(path_output);  
if (input.is_open()) {  
    for (char c; input.get(c);){  
        output.put(codebook[static_cast<unsigned char>(c)]);  
    }  
else{  
    std::cout << "Error Opening Input!" << std::endl;  
}  
}
```

注意:

1. 考虑文件打开出现异常的情况，使用 `is_open()`。
2. 使用 `codebook[static_cast<unsigned char>(c)]` 找到对应的密码。



# 纲要

---

- 第一部分：读取密码本
- 第二部分：编码
- 第三部分：译码

# 译码

```
std::string code_string;

code_string = read_codebook_char(path_codebook);
int n = code_string.length();
char codebook[n + 1];
for (int i = 0; i < n; i++){
    codebook[code_string[i] - 'a'] = 'a' + i;
}

std::ifstream input(path_input);
std::ofstream output(path_output);
if (input.is_open()) {
    for (char c; input.get(c);){
        if ((c > 'a') && (c <= 'z')){
            c = codebook[c - 'a'];
        }
        output.put(c);
    }
}
else{
    std::cout << "Error Opening Input!" << std::endl;
}
```

# 译码

```
std::vector<int> array;
array = read_codebook_number(path_codebook);

unsigned char codebook [256];
for (int i = 0; i < 256; ++i){
    codebook[static_cast<unsigned char>(array[i])] = i;
}

std::ifstream input(path_input);
std::ofstream output(path_output);
if (input.is_open()) {
    for (char c; input.get(c);){
        output.put(codebook[static_cast<unsigned char>(c)]);
    }
}
else{
    std::cout << "Error Opening Input!" << std::endl;
}
```

# 二进制读取

---

```
ifstream input (path_input, ios::in | ios::binary);
```

# 命令行操作

```
g++ encrypt.cpp utils.cpp -std=c++17 -o encrypt  
./encrypt codebook.txt input.txt output.txt
```

```
g++ decrypt.cpp utils.cpp -std=c++17 -o decrypt  
./decrypt codebook.txt input.txt output.txt
```

```
g++ encrypt_extend.cpp utils.cpp -std=c++17 -o encrypt_extend  
./encrypt_extend codebook.txt hello_world.o hello_world_encrypted.o
```

```
g++ decrypt_extend.cpp utils.cpp -std=c++17 -o decrypt_extend  
./decrypt_extend codebook.txt hello_world_encrypted.o hello_world.o
```

```
chmod +x ./hello_world_recovered.o (变成可执行文件)  
./hello_world_recovered.o
```



深蓝学院  
shenlanxueyuan.com

感谢各位聆听

Thanks for Listening

