

```
1  /*
2      資料結構HWK3字典
3      建構字典
4
5      架構
6          1. 建置資料庫
7          2. 目標直接搜尋
8
9  */
10 #include <iostream>
11 #include <fstream>
12
13 #define word_n 57
14
15 using namespace std;
16
17 struct words
18 {
19     string english_word, chinese_word;
20 };
21
22 struct word_table
23 {
24     int word_total = 0;
25     words word[200];
26 };
27
28 void read_date(word_table table_date[])
29 {
30     int id, ch_t = 0;
31     string in_english, in_chinese;
32     ifstream InF;
33     char FName[20], ch;
34
35     cout << "輸入檔名: ";
36     cin >> FName;
37
38     InF.open(FName, ios::in);
39
40     if (!InF)
41     {
42         cout << "檔案無法開啟\n";
43         exit(1);
44     }
45     else
46     {
47         while (InF.get(ch))
48         {
49             if (ch_t++ < 20) in_english += ch;
50             else if (ch_t++ >= 20 && ch != '\n') in_chinese += ch;
51             else if (ch == '\n')
52             {
53                 ch_t = 0;
```

```
54
55         id = in_english[0] * in_english[1] % word_n;
56         table_date[id].word[table_date[id].word_total].english_word =      ↗
            in_english;
57         table_date[id].word[table_date[id].word_total++].chinese_word =    ↗
            in_chinese;
58
59         //cout << in_english << " " << in_chinese << " id = " << id << endl;
60
61         in_english = "";
62         in_chinese = "";
63     }
64 }
65 InF.close();
66 }
67 }
68
69 void print_all(word_table table_date[])
70 {
71     for (int i = 0; i < word_n; i++)
72     {
73         cout << "i = " << i << endl;
74         for (int j = 0; j < table_date[i].word_total; j++)
75         {
76             cout << table_date[i].word[j].english_word;
77             cout << table_date[i].word[j].chinese_word << endl;
78         }
79         cout << endl;
80     }
81 }
82
83 void search(word_table table_date[], string search_word)
84 {
85     bool op = false;
86     int id = search_word[0] * search_word[1] % word_n;
87     for (int i = search_word.size(); i < 20; i++)
88     {
89         search_word += ' ';
90     }
91     for (int j = 0; j < table_date[id].word_total; j++)
92     {
93         if (table_date[id].word[j].english_word == search_word)
94         {
95             op = true;
96             cout << table_date[id].word[j].chinese_word << endl;
97             cout << "共搜尋" << j << "次" << endl;
98         }
99     }
100     if (op == true) cout << "搜尋完成" << endl;
101     else cout << "查無資料" << endl;
102 }
103
104 int main(void)
```

```
105 {
106     string search_word;
107     word_table table_date[word_n];
108
109     read_date(table_date);
110     //print_all(table_date);
111
112     bool op = true;
113     while (op != false)
114     {
115         cout << "請輸入要搜尋字串:" << endl;
116         cin >> search_word;
117         search(table_date, search_word);
118         cout << "是否再次使用(y/n)" << endl;
119         string next = "";
120         cin >> next;
121         if (next[0] == 'y' || next[0] == 'Y') op = true;
122         else op = false;
123     }
124 }
```