```cpp
/*
    資料結構HWK3字典
        建構字典

    架構
        1.建置資料庫
        2.目標直接搜尋

*/
#include <iostream>
#include <fstream>

#define word_n 57

using namespace std;

struct words
{
    string english_word, chinese_word;
};

struct word_table
{
    int word_total = 0;
    words word[200];
};

void read_date(word_table table_date[])
{
    int id, ch_t = 0;
    string in_english, in_chinese;
    ifstream InF;
    char FName[20], ch;

    cout << "輸入檔名: ";
    cin >> FName;

    InF.open(FName, ios::in);

    if (!InF)
    {
        cout << "檔案無法開啟\n";
        exit(1);
    }
    else
    {
        while (InF.get(ch))
        {
            if (ch_t++ < 20) in_english += ch;
            else if (ch_t++ >= 20 && ch != '\n') in_chinese += ch;
            else if (ch == '\n')
            {
                ch_t = 0;

                id = in_english[0] * in_english[1] % word_n;
                table_date[id].word[table_date[id].word_total].english_word = in_english;
                table_date[id].word[table_date[id].word_total++].chinese_word = in_chinese;

                //cout << in_english << " " << in_chinese << " id = " << id << endl;

                in_english = "";
                in_chinese = "";
            }
        }
        InF.close();
    }
}

void print_all(word_table table_date[])
{
    for (int i = 0; i < word_n; i++)
    {
        cout << "i = " << i << endl;
        for (int j = 0; j < table_date[i].word_total; j++)
        {
            cout << table_date[i].word[j].english_word;
            cout << table_date[i].word[j].chinese_word << endl;
        }
        cout << endl;
    }
}

void search(word_table table_date[], string search_word)
{
    bool op = false;
    int id = search_word[0] * search_word[1] % word_n;
    for (int i = search_word.size(); i < 20; i++)
    {
        search_word += ' ';
    }
    for (int j = 0; j < table_date[id].word_total; j++)
    {
        if (table_date[id].word[j].english_word == search_word)
        {
            op = true;
            cout << table_date[id].word[j].chinese_word << endl;
            cout << "共搜尋" << j << "次" << endl;
        }
    }
    if (op == true) cout << "搜尋完成" << endl;
    else cout << "查無資料" << endl;
}

int main(void)
```

```
{
    string search_word;
    word_table table_date[word_n];

    read_date(table_date);
    //print_all(table_date);

    bool op = true;
    while (op != false)
    {
        cout << "請輸入要搜尋字串:" << endl;
        cin >> search_word;
        search(table_date, search_word);
        cout << "是否再次使用(y/n)" << endl;
        string next = "";
        cin >> next;
        if (next[0] == 'y' || next[0] == 'Y') op = true;
        else op = false;
    }
}
```