

```
1  /*
2      hwk2
3      name    LU,YUDA
4      number  b10702057
5      date    20.05.23(1.0), 20,05,24(1.1) 20.05.24(2.0)
6  */
7
8  /*
9      大架構
10     1.使用者選擇排序方式
11         a. 選擇資料結構
12     2.使用者輸入檔案名稱
13         a. 讀黨
14         b. 寫入檔案
15     3.排序
16         a. 依照使用者輸入排序法執行
17     4.輸出最終答案
18         a. 輸出成檔案
19  */
20
21 #include <iostream>
22 #include <fstream>
23 #include <stdio.h>
24 #include <stdlib.h>
25 #include <iomanip>
26 #include <windows.h>
27 #include <dos.h>
28 #include <conio.h>
29 using namespace std;
30
31 /*
32     sort_name 存放要使用哪一種排序方式
33  */
34
35 int sort_name = -1;
36
37 int getMax(int array[], int n) {
38     int max = array[0];
39     for (int i = 1; i < n; i++)
40         if (array[i] > max)
41             max = array[i];
42     return max;
43 }
44
45 void countingSort(int array[], int size, int place)
46 {
47     /*
48         架構
49         place        >> 排序位元數
50         output[1000] >> 輸出暫存
51         count[max]   >> max十進位
52
53         1.計算出每一個位元有多少個
```

```
54         ex: 11 % 10 = 1 有幾個
55         2.讓count跟陣列位子對齊
56         3.校正到output
57         4.寫入原本array
58
59     */
60
61     const int max = 10;
62     int output[1000];
63     int count[max];
64
65     for (int i = 0; i < max; ++i)
66     {
67         count[i] = 0;
68     }
69
70     for (int i = 0; i < size; i++)
71     {
72         count[(array[i] / place) % 10]++;
73     }
74
75     for (int i = 1; i < max; i++)
76     {
77         count[i] += count[i - 1];
78     }
79
80     for (int i = size - 1; i >= 0; i--)
81     {
82         output[count[(array[i] / place) % 10] - 1] = array[i];
83         count[(array[i] / place) % 10]--;
84     }
85
86     for (int i = 0; i < size; i++)
87     {
88         array[i] = output[i];
89     }
90 }
91
92 void Radix(int number_array[])
93 {
94     int size = 1000;
95     /*
96         getMax 先找最大值
97         countingSort
98         利用目前位元排序
99     */
100     int max = getMax(number_array, size);
101     for (int place = 1; max / place > 0; place *= 10)
102     {
103         countingSort(number_array, size, place);
104     }
105 }
106
```

```
107 int Shell(int number_array[])
108 {
109     /*
110         雙迴圈設計
111         外迴圈
112         gap 控制 間個差距
113         內迴圈
114         i 右手 從gap那個位子開始
115         j 左手 從初始位置開始
116         如果左手大於右手
117         swap(i, j)
118     */
119
120     int size = 1000;
121     for (int gap = size / 2; gap > 0; gap /= 2)
122     {
123         for (int i = gap; i < size; i += 1)
124         {
125             int temp = number_array[i];
126
127             int j;
128             for (j = i; j >= gap && number_array[j - gap] > temp; j -= gap)
129             {
130                 number_array[j] = number_array[j - gap];
131             }
132             number_array[j] = temp;
133         }
134     }
135     return 0;
136 }
137
138 void outF(int number_array[])
139 {
140     FILE* pFile;
141     char f_name[30];
142
143     cout << "輸入檔名: ";
144     cin >> f_name;
145
146     pFile = fopen(f_name, "w");
147
148     if (NULL == pFile)
149     {
150         printf("open failure");
151     }
152     else
153     {
154         for (int i = 0; i < 1000; i++) fprintf(pFile, "%5d\n", number_array[i]);
155     }
156     fclose(pFile);
157
158     cout << "打印到.txt完畢\n";
159 }
```

```
160
161 void read_date(int number_array[])
162 {
163     ifstream InF;
164     int n = 0, i = 0;
165     char FName[20], ch[10];
166     string in_s;
167
168     cout << "輸入檔名: ";
169     cin >> FName;
170
171     InF.open(FName, ios::in);
172
173     if (!InF)
174     {
175         cout << "檔案無法開啟\n";
176         exit(1);
177     }
178     else
179     {
180         while (InF.getline(ch, 10))
181         {
182             if (ch[0] != '\n')
183             {
184                 in_s = ch;
185                 number_array[i++] = atoi(in_s.c_str());
186             }
187         }
188         InF.close();
189     }
190 }
191
192 int main(void)
193 {
194     cout << "1.Radix sort 2,Shell sort: ";
195     cin >> sort_name;
196
197     int number_array[1010];
198     if (sort_name == 1 || sort_name == 2)
199     {
200         read_date(number_array);
201         if (sort_name == 1) Radix(number_array);
202         else Shell(number_array);
203     }
204     else
205     {
206         cout << "輸入錯誤!!" << endl;
207         exit(1);
208     }
209
210     outF(number_array);
211     system("pause");
212 }
```