

程式設計 HW6 Part3 Coding

108072147 林汶螢

1.

```
1.cpp
1.cpp > main(int, char * [])
4
5 bool strcmpare(string a, string b){
6     //return true, if a < b
7     for(int i=0; i< min(a.length(), b.length()); i++){
8         if(a[i] < b[i]) return true;
9         else if(a[i] > b[i])return false;
10    }
11    return ( a.length() < b.length() );
12 }
13
14 void selectionSort(string array[], int size){
15     int start, minIndex;
16     string minValue;
17     for (start = 0; start < (size - 1); start++){
18         minIndex = start;
19         minValue = array[start];
20         for (int index = start + 1; index < size; index++){
21             if (strcmpare(array[index], minValue)){
22                 minValue = array[index];
23                 minIndex = index;
24             }
25         }
26         array[minIndex] = array[start];
27         array[start] = minValue;
28     }
29 }
30
31 int main(int argc, char* argv[]){
32     const int SIZE = 20;
33     string name[SIZE] =
34     {"Collins, Bill", "Smith, Bart", "Michalski, Joe", "Griffin, Jim",
35     "Sanchez, Manny", "Rubin, Sarah", "Taylor, Tyrone", "Johnson, Jill",
36     "Allison, Jeff", "Moreno, Juan", "Wolfe, Bill", "Whitman, Jean",
37     "Moretti, Bella", "Wu, Hong", "Patel, Renee", "Harrison, Rose",
38     "Smith, Cathy", "Conroy, Pat", "Kelly, Sean", "Holland, Beth"};
39     selectionSort(name, SIZE);
40     for (int i=0; i<SIZE; i++){
41         cout<<setw(2)<<right<<i<<". "<<name[i]<<endl;
42     }
43     return 0;
44 }
```

```
PS C:\Users\User\OneDrive\Introduction-to-programming\C++HW6> g++ .\1.cpp -o 1.exe
PS C:\Users\User\OneDrive\Introduction-to-programming\C++HW6> .\1.exe
0. Allison, Jeff
1. Collins, Bill
2. Conroy, Pat
3. Griffin, Jim
4. Harrison, Rose
5. Holland, Beth
6. Johnson, Jill
7. Kelly, Sean
8. Michalski, Joe
9. Moreno, Juan
10. Moretti, Bella
11. Patel, Renee
12. Rubin, Sarah
13. Sanchez, Manny
14. Smith, Bart
15. Smith, Cathy
16. Taylor, Tyrone
17. Whitman, Jean
18. Wolfe, Bill
19. Wu, Hong
PS C:\Users\User\OneDrive\Introduction-to-programming\C++HW6> |
```

2.

```
2.cpp > selectionSort(vector<string> &)\n1  #include <iostream>\n2  #include <fstream>\n3  #include <iomanip>\n4  #include <vector>\n5  using namespace std;\n6\n7  bool strcmpare(string a, string b){\n8      //return true, if a < b\n9      for(int i=0; i< min(a.length(), b.length()); i++){ \n10         if(a[i] < b[i]) return true;\n11         else if(a[i] > b[i])return false;\n12     }\n13     return ( a.length() < b.length() );\n14 }\n15\n16 void selectionSort( vector <string> &array){\n17     int start, minIndex;\n18     string minValue;\n19     for (start = 0; start < array.size() -1; start++){ \n20         minIndex = start;\n21         minValue = array[start];\n22         for (int index = start + 1; index < array.size(); index++){ \n23             if (strcmpare(array[index], minValue)){\n24                 minValue = array[index];\n25                 minIndex = index;\n26             }\n27         }\n28         array[minIndex] = array[start];\n29         array[start] = minValue;\n30     }\n31 }\n32\n33 int main(int argc, char* argv){\n34     vector <string> name;\n35     fstream fnames;\n36     string temp;\n37     fnames.open("names.dat",ios::in);\n38     while(getline(fnames,temp)){\n39         name.push_back(temp);\n40     }\n41     fnames.close();\n42\n43     selectionSort(name);\n44     for (int i=0; i< name.size() ; i++){ \n45         cout<<setw(2)<<right<<i<<". "<<name[i]<<endl;\n46     }\n47     return 0;\n48 }\n49
```

```
names.dat\n1  Collins, Bill\n2  Smith, Bart\n3  Michalski, Joe\n4  Griffin, Jim\n5  Sanchez, Manny\n6  Rubin, Sarah\n7  Taylor, Tyrone\n8  Johnson, Jill\n9  Allison, Jeff\n10 Moreno, Juan\n11 Wolfe, Bill\n12 Whitman, Jean\n13 Moretti, Bella\n14 Wu, Hong\n15 Patel, Renee\n16 Harrison, Rose\n17 Smith, Cathy\n18 Conroy, Pat\n19 Kelly, Sean\n20 Holland, Beth
```

此為名單檔案

執行結果

```
0. Allison, Jeff\n1. Collins, Bill\n2. Conroy, Pat\n3. Griffin, Jim\n4. Harrison, Rose\n5. Holland, Beth\n6. Johnson, Jill\n7. Kelly, Sean\n8. Michalski, Joe\n9. Moreno, Juan\n10. Moretti, Bella\n11. Patel, Renee\n12. Rubin, Sarah\n13. Sanchez, Manny\n14. Smith, Bart\n15. Smith, Cathy\n16. Taylor, Tyrone\n17. Whitman, Jean\n18. Wolfe, Bill\n19. Wu, Hong
```

3.

```
G++ 3.cpp > strcompare(string, string)
1  #include <iostream>
2  using namespace std;
3
4  bool strcompare(string a, string b){
5      //return true, if a < b
6      for(int i=0; i< min(a.length(), b.length()); i++){
7          if(a[i] < b[i]) return true;
8          else if(a[i] > b[i])return false;
9      }
10     return ( a.length() < b.length() );
11 }
12
13 int binarySearch(const string array[], int size, string value){
14     int first = 0,
15         last = size - 1,
16         middle,
17         position = -1;
18     bool found = false;
19     while (!found && first <= last){
20         middle = (first + last) / 2;
21         if (array[middle] == value)
22         {
23             found = true;
24             position = middle;
25         }
26         else if (strcompare(value, array[middle]))
27             last = middle - 1;
28         else
29             first = middle + 1;
30     }
31     return position;
32 }
33
34 int main(int argc, char* argv[]){
35     const int SIZE = 20;
36     int ans;
37     string search;
38     string name[SIZE] ={
39         "Allison, Jeff","Collins, Bill","Conroy, Pat","Griffin, Jim",
40         "Harrison, Rose","Holland, Beth","Johnson, Jill","Kelly, Sean",
41         "Michalski, Joe","Moreno, Juan","Moretti, Bella","Patel, Renee",
42         "Rubin, Sarah","Sanchez, Manny","Smith, Bart","Smith, Cathy",
43         "Taylor, Tyrone","Whitman, Jean","Wolfe, Bill","Wu, Hong"    };
44
45     cout<<"Please enter the name u want to search: ";
46     getline(cin,search);
47     ans = binarySearch(name, SIZE, search);
48     if(ans >= 0) cout<<"The position of ur request is : "<< ans <<endl;
49     else cout<<"Failed to find ur request."<<endl;
50
51     return 0;
52 }
```

此為有找到之結果

```
Please enter the name u want to search: Johnson, Jill
The position of ur request is : 6
```

此為沒找到之結果

```
Please enter the name u want to search: Angry
Failed to find ur request.
```

4.

```
G++ 4.cpp > bubbleSort(int [], int)
1  #include <iostream>
2  #include <iomanip>
3  #include <ctime>
4  #include <cstdlib>
5
6  using namespace std;
7  int bubbleSort(int array[], int size){
8      int temp, counter=0;
9      bool swap;
10     do{
11         swap = false;
12         for (int count = 0; count < (size - 1); count++){
13             if (array[count] > array[count + 1]){
14                 temp = array[count];
15                 array[count] = array[count + 1];
16                 array[count + 1] = temp;
17                 swap = true;
18                 counter++;
19             }
20         }
21     } while (swap);
22     return counter;
23 }
24
25 int selectionSort(int array[], int size){
26     int start, minIndex, minValue, counter=0;
27     for (start = 0; start < (size - 1); start++){
28         minIndex = start;
29         minValue = array[start];
30         for (int index = start + 1; index < size; index++){
31             if (array[index] < minValue){
32                 minValue = array[index];
33                 minIndex = index;
34                 counter++;
35             }
36         }
37         array[minIndex] = array[start];
38         array[start] = minValue;
39     }
40     return counter;
41 }
42
43 int main(int argc, char* argv){
44     srand(time(NULL));
45     const int MAX=30;
46     int sum1=0, sum2=0;
47     int random1[MAX]={0}, random2[MAX]={0};
48     for(int i=0; i<20; i++){
49         for(int j=0; j<MAX; j++){
50             random1[j] = rand();
51             random2[j] = random1[j];
52         }
53         int count1 = selectionSort( random1, MAX);
54         sum1 += count1;
55         int count2 = bubbleSort( random2, MAX);
56         sum2 += count2;
57         cout<<setw(2)<<i<<" Selection Sort : "
58             <<count1<<"/ Bubble Sort : "<<count2<<endl;
59     }
60     cout<<setfill(' ')<<setw(17)<<right<<"Average"<<setw(10)<<"\n";
61     cout<<setfill(' ')<<"Selection Sort : "<<setw(8)<<right<<sum1/20<<endl
62         <<"Bubble Sort : "<<setw(12)<<right<<sum2/20<<endl;
63     return 0;
64 }
```

此為執行結果

```
PS C:\Users\User\OneDrive\Introduction-to-programming\C++HW6> g++ .\4.cpp -o 4.exe
PS C:\Users\User\OneDrive\Introduction-to-programming\C++HW6> .\4.exe
0. Selection Sort : 56/ Bubble Sort : 162
1. Selection Sort : 59/ Bubble Sort : 206
2. Selection Sort : 75/ Bubble Sort : 255
3. Selection Sort : 62/ Bubble Sort : 207
4. Selection Sort : 68/ Bubble Sort : 210
5. Selection Sort : 67/ Bubble Sort : 247
6. Selection Sort : 52/ Bubble Sort : 239
7. Selection Sort : 65/ Bubble Sort : 225
8. Selection Sort : 69/ Bubble Sort : 228
9. Selection Sort : 62/ Bubble Sort : 211
10. Selection Sort : 64/ Bubble Sort : 218
11. Selection Sort : 72/ Bubble Sort : 262
12. Selection Sort : 80/ Bubble Sort : 238
12. Selection Sort : 80/ Bubble Sort : 238
13. Selection Sort : 57/ Bubble Sort : 220
14. Selection Sort : 65/ Bubble Sort : 237
15. Selection Sort : 75/ Bubble Sort : 215
16. Selection Sort : 53/ Bubble Sort : 222
17. Selection Sort : 80/ Bubble Sort : 218
18. Selection Sort : 76/ Bubble Sort : 219
19. Selection Sort : 70/ Bubble Sort : 249
=====Average=====
Selection Sort :      66
Bubble Sort :      224
PS C:\Users\User\OneDrive\Introduction-to-programming\C++HW6>
```

5.

```

5.cpp > linearSearch(const int [], int, int)
1  #include <iostream>
2  #include <iomanip>
3  #include <ctime>
4  #include <cstdlib>
5  #include <algorithm>
6  using namespace std;
7
8  int linearSearch(const int array[], int size, int value){
9      int index = 0;
10     int position = -1;
11     int counter = 0;
12     bool found = false;
13     while (index < size && !found){
14         if (array[index] == value){
15             found = true;
16             position = index;
17         }
18         index++;
19         counter++;
20     }
21     return counter;
22 }
23
24 int binarySearch(const int array[], int size, int value){
25     int first = 0, last = size - 1, middle, position = -1;
26     int counter=0;
27     bool found = false;
28     while (!found && first <= last){
29         middle = (first + last) / 2;
30         if (array[middle] == value){
31             found = true;
32             position = middle;
33         }
34         else if (array[middle] > value)
35             last = middle - 1;
36         else
37             first = middle + 1;
38         counter++;
39     }
40     return counter;
41 }
42
43 int main(int argc, char* argv[]){
44     srand(time(NULL));
45     const int MAX=30;
46     int sum1=0, sum2=0, guess=0;
47     int random1[MAX]={0};
48     for(int i=0; i<20; i++){
49         for(int j=0; j<MAX; j++){
50             random1[j] = rand();
51         }
52         guess = random1[rand()%MAX];
53         sort( random1, random1+MAX );
54         int count1 = linearSearch( random1, MAX, guess);
55         sum1 += count1;
56         int count2 = binarySearch( random1, MAX, guess);
57         sum2 += count2;
58         cout<<setw(2)<<i<<" Linear Search : "
59              <<count1<<"/ Binary Search : "<<count2<<endl;
60     }
61     cout<<setfill(' ')<<setw(17)<<right<<"Average"<<setw(10)<<"\n";
62     cout<<setfill(' ')<<"Linear Search : "<<setw(10)<<right<<sum1/20<<endl;
63     cout<<setfill(' ')<<"Binary Search : "<<setw(10)<<right<<sum2/20<<endl;
64     return 0;
65 }

```

```

PS C:\Users\User\OneDrive\Introduction-to-programming\C++HW6> ./5.exe
0. Linear Search : 14/ Binary Search : 5
1. Linear Search : 26/ Binary Search : 5
2. Linear Search : 5/ Binary Search : 4
3. Linear Search : 22/ Binary Search : 5
4. Linear Search : 26/ Binary Search : 5
5. Linear Search : 23/ Binary Search : 2
6. Linear Search : 3/ Binary Search : 3
7. Linear Search : 8/ Binary Search : 5
8. Linear Search : 21/ Binary Search : 4
9. Linear Search : 18/ Binary Search : 5
10. Linear Search : 19/ Binary Search : 3
11. Linear Search : 17/ Binary Search : 4
12. Linear Search : 5/ Binary Search : 4
13. Linear Search : 14/ Binary Search : 5
14. Linear Search : 19/ Binary Search : 3
15. Linear Search : 25/ Binary Search : 4
16. Linear Search : 4/ Binary Search : 5
17. Linear Search : 15/ Binary Search : 1
18. Linear Search : 13/ Binary Search : 4
19. Linear Search : 15/ Binary Search : 1
=====Average=====
Linear Search :          15
Binary Search :          3

```

6.

```

C:\cpp> ...
1 #include <iostream>
2 #include <sstream>
3 using namespace std;
4
5 class Stats{
6 private:
7     string city;
8     string monthYear;
9     double rainfall[30];
10    int cnt;
11    int most, least;
12    const string month[12]={ "January", "February", "March",
13                             "April", "May", "June", "July",
14                             "August", "September", "October",
15                             "November", "December"
16                             };
17 public:
18     Stats(string city, string monthYear){
19         this->city = city;
20         this->monthYear = monthYear;
21         for(int i=0; i<30; i++) this->rainfall[i] = 0;
22         this->cnt = 0;
23         this->most = 0;
24         this->least = 0;
25     }
26     double total(){
27         double sum = 0;
28         for(int i=0; i< this->cnt; i++){
29             sum += this->rainfall[i];
30         }
31         return sum;
32     }
33     double average(){
34         return this->total()/this->cnt;
35     }
36     double lowest(){
37         double L=rainfall[0];
38         for(int i=0; i< this->cnt; i++){
39             if(this->rainfall[i] < L){
40                 least = i;
41                 L = this->rainfall[i];
42             }
43         }
44         return L;
45     }
46     double highest(){
47         double M=rainfall[0];
48         for(int i=0; i< this->cnt; i++){
49             if(this->rainfall[i] > M){
50                 most = i;
51                 M = this->rainfall[i];
52             }
53         }
54         return M;
55     }
56     bool storeValue(double rainfall){
57         if (rainfall<0 || cnt>30) return false;
58         this->rainfall[this->cnt] = rainfall;
59         cnt++;
60         return true;
61     }
62     string convert(){
63         stringstream ss;
64         int monthnum;
65
66         ss << this->monthYear.substr(4,2);
67         ss >> monthnum;
68         return this->month[monthnum] + ", " + this->monthYear.substr(0,4);
69     }
70
71     string convert(int cnt){
72         stringstream ss;
73         int monthnum, yearnum;
74         string yearstr;
75
76         ss << this->monthYear.substr(4,2);
77         ss >> monthnum;
78         ss.clear();
79
80         ss << this->monthYear.substr(0,4);
81         ss >> yearnum;
82         ss.clear();
83
84         yearnum += (monthnum + cnt)/12;
85         monthnum = (monthnum + cnt)%12;
86
87         ss << yearnum;
88         ss >> yearstr;
89
90         return this->month[monthnum] + ", " + yearstr;
91     }
92     void selectionSort(int array[], int size, int cnt){

```

```

P5 C:\Users\User\OneDrive\Introduction-to-programming\C++HW6> ./6.exe
Please enter the city u want to analyze:TPE
Please enter the month u started to analyze:201801

```

```

How's the rainfall in TPE ?
100
Do u want to continue?
Please enter Y/N only.
Y

```

```

How's the rainfall in TPE ?
10
Do u want to continue?
Please enter Y/N only.
Y
Do u want to continue?
Please enter Y/N only.
Y

```

```

How's the rainfall in TPE ?
15
Do u want to continue?
Please enter Y/N only.
Y

```

```

How's the rainfall in TPE ?
645
Do u want to continue?
Please enter Y/N only.
Y

```

```

How's the rainfall in TPE ?
799
Do u want to continue?
Please enter Y/N only.
Y

```

```

How's the rainfall in TPE ?
657
Do u want to continue?
Please enter Y/N only.
N

```

Rainfall Report Display by ascending in TPE County :

```

-----
March, 2018 Rainfall: 10
April, 2018 Rainfall: 15
February, 2018 Rainfall: 100
May, 2018 Rainfall: 645
July, 2018 Rainfall: 657
June, 2018 Rainfall: 799

```

Rainfall Report Display by descending in TPE County :

```

-----
June, 2018 Rainfall: 799
July, 2018 Rainfall: 657
May, 2018 Rainfall: 645
February, 2018 Rainfall: 100
April, 2018 Rainfall: 15
March, 2018 Rainfall: 10

```

Rainfall Report Display by time in TPE County :

```

-----
February, 2018 Rainfall: 100
March, 2018 Rainfall: 10
April, 2018 Rainfall: 15
May, 2018 Rainfall: 645
June, 2018 Rainfall: 799
July, 2018 Rainfall: 657

```

6.cpp > _

```
void selectionSort(int array[], int size, int cnt[]){
93     int start, minIndex, minValue, minCNT;
94     for (start = 0; start < (size - 1); start++){
95         minIndex = start;
96         minValue = array[start];
97         minCNT = cnt[start];
98         for (int index = start + 1; index < size; index++){
99             if (array[index] < minValue){
100                 minValue = array[index];
101                 minIndex = index;
102                 minCNT = cnt[index];
103             }
104         }
105         array[minIndex] = array[start];
106         array[start] = minValue;
107         cnt[minIndex] = cnt[start];
108         cnt[start] = minCNT;
109     }
110 }

string displayReport(){
111     string ret;
112     ret += this->convert() + " - " + this->convert(cnt) + " Rain Report for " + this->city + " County\n";
113     stringstream temp;
114     string strtemp;
115     temp<< total();
116     temp>> strtemp;
117     temp.clear();
118     ret += "Total rainfall in this period: " + strtemp + " inches\n";
119     temp<< average();
120     temp>> strtemp;
121     temp.clear();
122     ret += "Average monthly rainfall: " + strtemp + " inches\n";
123     temp<< lowest();
124     temp>> strtemp;
125     temp.clear();
126     ret += "The least rain fell in " + this->convert(lowest) + " with " + strtemp + " inches\n";
127     temp<< highest();
128     temp>> strtemp;
129     temp.clear();
130     ret += "The most rain fell in " + this->convert(most) + " with " + strtemp + " inches\n";
131     return ret;
132 }

string displayAscending(){
133     string ret;
134     ret += " Rainfall Report Display by ascending in " + this->city + " County : \n";
135     ret += "-----\n";
136     int copyRainfall[this->cnt], cnt[this->cnt];
137     for(int i=0; i<this->cnt; i++){
138         copyRainfall[i] = this->rainfall[i];
139         cnt[i] = i;
140     }
141     selectionSort(copyRainfall, this->cnt, cnt);
142     for(int i=0; i<this->cnt; i++){
143         stringstream temp;
144         string strtemp;
145         temp<< copyRainfall[i];
146         temp>> strtemp;
147         temp.clear();
148         ret += convert(cnt[i]) + " Rainfall: " + strtemp + "\n";
149     }
150     return ret;
151 }

string displayDescending(){
152     string ret;
153     ret += " Rainfall Report Display by descending in " + this->city + " County : \n";
154     ret += "-----\n";
155     int copyRainfall[this->cnt], cnt[this->cnt];
156     for(int i=0; i<this->cnt; i++){
157         copyRainfall[i] = this->rainfall[i];
158         cnt[i] = i;
159     }
160     selectionSort(copyRainfall, this->cnt, cnt);
161     for(int i=this->cnt-1; i >= 0; i--){
162         stringstream temp;
163         string strtemp;
164         temp<< copyRainfall[i];
165         temp>> strtemp;
166         temp.clear();
167         ret += convert(cnt[i]) + " Rainfall: " + strtemp + "\n";
168     }
169     return ret;
170 }

string displayTime(){
171     string ret;
172     ret += " Rainfall Report Display by time in " + this->city + " County : \n";
173     ret += "-----\n";
174     for(int i=0; i<this->cnt; i++){
175         stringstream temp;
176         string strtemp;
177         temp<< rainfall[i];
178         temp>> strtemp;
179     }
180 }
```


G: 6.cpp > ...

```
184         temp.clear();
185         ret += convert(i)+" Rainfall: "+ strtemp +"\n";
186     }
187     return ret;
188 }
189 };
190 int main (int argc, char* argv[]){
191     string city,monthYear;
192     double rainfall;
193     int cont;
194     string s;
195     char c;
196
197     cout<<"Please enter the city u want to analyze:";
198     getline(cin,city);
199     cout<<"Please enter the month u started to analyze:";
200     cin>> monthYear;
201     Stats stats(city, monthYear);
202
203     cont = 1;
204     while(cont){
205         do{
206             cout<<"\nHow's the rainfall in "<<city<<" ?\n";
207             cin>>rainfall;
208         }while(!stats.storeValue(rainfall));
209         getline(cin,s);
210
211         do{
212             cout<<"Do u want to continue?\n";
213             cout<<"Please enter Y/N only.\n";
214             getline(cin,s);
215             if(s[0]=='Y') cont =1;
216             else cont = 0;
217         }while(s.size()>1 || (s[0] != 'Y' && s[0] != 'N'));
218     }
219
220     cout<<stats.displayAscending()<<endl;
221     cout<<stats.displayDescending()<<endl;
222     cout<<stats.displayTime()<<endl;
223     return 0;
224 }
```