程式設計 HW6 Part3 Coding

108072147 林汶螢

1.

```cpp
bool strcompare(string a, string b){
    //return true, if a < b
    for(int i=0; i< min(a.length(), b.length()); i++){
       if(a[i] < b[i]) return true;
       else if(a[i] > b[i])return false;
    }
    return ( a.length() < b.length() );
}

void selectionSort(string array[], int size){
    int     start, minIndex;
    string  minValue;
    for (start = 0; start < (size - 1); start++){
        minIndex = start;
        minValue = array[start];
        for (int index = start + 1; index < size; index++){
            if (strcompare(array[index], minValue)){
                minValue = array[index];
                minIndex = index;
            }
        }
        array[minIndex] = array[start];
        array[start] = minValue;
    }
}

int main(int argc, char* argv[]){
    const int SIZE = 20;
    string name[SIZE] =
    {"Collins, Bill", "Smith, Bart", "Michalski, Joe", "Griffin, Jim",
    "Sanchez, Manny", "Rubin, Sarah", "Taylor, Tyrone", "Johnson, Jill",
    "Allison, Jeff", "Moreno, Juan", "Wolfe, Bill", "Whitman, Jean",
    "Moretti, Bella", "Wu, Hong", "Patel, Renee", "Harrison, Rose",
    "Smith, Cathy", "Conroy, Pat", "Kelly, Sean", "Holland, Beth"};
    selectionSort(name, SIZE);
    for (int i=0; i<SIZE; i++){
        cout<<setw(2)<<right<<i<<". "<<name[i]<<endl;
    }
    return 0;
}
```

```
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>&)

```cpp
#include <iostream>
#include <fstream>
#include <iomanip>
#include <vector>
using namespace std;

bool strcompare(string a, string b){
    //return true, if a < b
    for(int i=0; i< min(a.length(), b.length()); i++){
       if(a[i] < b[i]) return true;
       else if(a[i] > b[i])return false;
    }
    return ( a.length() < b.length() );
}

void selectionSort( vector <string> &array){
    int     start, minIndex;
    string  minValue;
    for (start = 0; start < array.size() -1; start++){
        minIndex = start;
        minValue = array[start];
        for (int index = start + 1; index < array.size(); index++){
            if (strcompare(array[index], minValue)){
                minValue = array[index];
                minIndex = index;
            }
        }
        array[minIndex] = array[start];
        array[start] = minValue;
    }
}

int main(int argc, char* argv[]){
    vector <string> name;
    fstream fnames;
    string temp;
    fnames.open("names.dat",ios::in);
    while(getline(fnames,temp)){
        name.push_back(temp);
    }
    fnames.close();

    selectionSort(name);
    for (int i=0; i< name.size() ; i++){
        cout<<setw(2)<<right<<i<<". "<<name[i]<<endl;
    }
    return 0;
}
```

names.dat

```
Collins, Bill
Smith, Bart
Michalski, Joe
Griffin, Jim
Sanchez, Manny
Rubin, Sarah
Taylor, Tyrone
Johnson, Jill
Allison, Jeff
Moreno, Juan
Wolfe, Bill
Whitman, Jean
Moretti, Bella
Wu, Hong
Patel, Renee
Harrison, Rose
Smith, Cathy
Conroy, Pat
Kelly, Sean
Holland, Beth
```

此為名單檔案

執行結果

```
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
```

3.

3.cpp > strcompare(string, string)

```cpp
#include <iostream>
using namespace std;

bool strcompare(string a, string b){
    //return true, if a < b
    for(int i=0; i< min(a.length(), b.length()); i++){
       if(a[i] < b[i]) return true;
       else if(a[i] > b[i])return false;
    }
    return ( a.length() < b.length() );
}

int binarySearch(const string array[], int size, string value){
    int first = 0,
        last = size - 1,
        middle,
        position = -1;
    bool found = false;
    while (!found && first <= last){
        middle = (first + last) / 2;
        if (array[middle] == value)
        {
            found = true;
            position = middle;
        }
        else if (strcompare(value, array[middle]))
            last = middle - 1;
        else
            first = middle + 1;
    }
    return position;
}

int main(int argc, char* argv[]){
    const int SIZE = 20;
    int ans;
    string search;
    string name[SIZE] ={
    "Allison, Jeff","Collins, Bill","Conroy, Pat","Griffin, Jim",
    "Harrison, Rose","Holland, Beth","Johnson, Jill","Kelly, Sean",
    "Michalski, Joe","Moreno, Juan","Moretti, Bella","Patel, Renee",
    "Rubin, Sarah","Sanchez, Manny","Smith, Bart","Smith, Cathy",
    "Taylor, Tyrone","Whitman, Jean","Wolfe, Bill","Wu, Hong"        };

    cout<<"Please enter the name u want to search: ";
    getline(cin,search);
    ans = binarySearch(name, SIZE, search);
    if(ans >= 0) cout<<"The position of ur request is : "<< ans <<endl;
    else cout<<"Failed to find ur request."<<endl;

    return 0;
}
```

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.

4.cpp > bubbleSort(int [], int)

```cpp
#include <iostream>
#include <iomanip>
#include <ctime>
#include <cstdlib>

using namespace std;
int bubbleSort(int array[], int size){
    int temp, counter=0;
    bool swap;
    do{
        swap = false;
        for (int count = 0; count < (size - 1); count++){
            if (array[count] > array[count + 1]){
                temp = array[count];
                array[count] = array[count + 1];
                array[count + 1] = temp;
                swap = true;
                counter++;
            }
        }
    } while (swap);
    return counter;
}

int selectionSort(int array[], int size){
    int start, minIndex, minValue, counter=0;
    for (start = 0; start < (size - 1); start++){
        minIndex = start;
        minValue = array[start];
        for (int index = start + 1; index < size; index++){
            if (array[index] < minValue){
                minValue = array[index];
                minIndex = index;
                counter++;
            }
        }
        array[minIndex] = array[start];
        array[start] = minValue;
    }
    return counter;
}

int main(int argc,char* argv[]){
    srand(time(NULL));
    const int MAX=30;
    int sum1=0,sum2=0;
    int random1[MAX]={0},random2[MAX]={0};
    for(int i=0; i<20; i++){
        for(int j=0; j<MAX; j++){
            random1[j] = rand();
            random2[j] = random1[j];
        }
        int count1 = selectionSort( random1, MAX);
        sum1 += count1;
        int count2 = bubbleSort( random2, MAX);
        sum2 += count2;
        cout<<setw(2)<<i<<". Sellection Sort : "
            <<count1<<"/ Bubble Sort : "<<count2<<endl;
    }
    cout<<setfill('=')<<setw(17)<<right<<"Average"<<setw(10)<<"\n";
    cout<<setfill(' ')<<"Sellection Sort : "<<setw(8)<<right<<sum1/20<<endl
        <<"Bubble Sort : "<<setw(12)<<right<<sum2/20<<endl;
    return 0;
}
```

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

5.

5.cpp > linearSearch(const int [], int, int)

```cpp
#include <iostream>
#include <iomanip>
#include <ctime>
#include <cstdlib>
#include <algorithm>
using namespace std;

int linearSearch(const int array[], int size, int value){
    int index = 0;
    int position = -1;
    int counter = 0;
    bool found = false;
    while (index < size && !found){
        if (array[index] == value){
            found = true;
            position = index;
        }
        index++;
        counter++;
    }
    return counter;
}

int binarySearch(const int array[], int size, int value){
    int first = 0, last = size - 1, middle, position = -1;
    int counter=0;
    bool found = false;
    while (!found && first <= last){
        middle = (first + last) / 2;
        if (array[middle] == value){
            found = true;
            position = middle;
        }
        else if (array[middle] > value)
            last = middle - 1;
        else
            first = middle + 1;
        counter++;
    }
    return counter;
}

int main(int argc,char* argv[]){
    srand(time(NULL));
    const int MAX=30;
    int sum1=0,sum2=0,guess=0;
    int random1[MAX]={0};
    for(int i=0; i<20; i++){
        for(int j=0; j<MAX; j++){
            random1[j] = rand();
        }
        guess = random1[rand()%MAX];
        sort( random1, random1+MAX );
        int count1 = linearSearch( random1, MAX, guess);
        sum1 += count1;
        int count2 = binarySearch( random1, MAX, guess);
        sum2 += count2;
        cout<<setw(2)<<i<<". Linear Search : "
            <<count1<<"/ Binary Search : "<<count2<<endl;
    }
    cout<<setfill('=')<<setw(17)<<right<<"Average"<<setw(10)<<"\n";
    cout<<setfill(' ')<<"Linear Search : "<<setw(10)<<right<<sum1/20<<endl
        <<"Binary Search : "<<setw(10)<<right<<sum2/20<<endl;
    return 0;
}
```

```
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.

```cpp
#include <iostream>
#include <sstream>
using namespace std;

class Stats{
    private:
        string city;
        string monthYear;
        double rainfall[30];
        int cnt;
        int most, least;
        const string month[12]={  "January", "February", "March",
                                  "April", "May", "June", "July",
                                  "August","September","October",
                                  "November","December"
                                };
    public:
        Stats(string city, string monthYear){
            this -> city = city;
            this -> monthYear = monthYear;
            for(int i=0; i<30; i++) this -> rainfall[i] = 0;
            this -> cnt = 0;
            this -> most = 0;
            this -> least = 0;
        }
        double total(){
            double sum = 0;
            for(int i=0; i< this->cnt; i++){
                sum += this->rainfall[i];
            }
            return sum;
        }
        double average(){
            return this->total()/this->cnt;
        }
        double lowest(){
            double L=rainfall[0];
            for(int i=0; i< this->cnt; i++){
                if(this->rainfall[i] < L){
                    least = i;
                    L = this->rainfall[i];
                }
            }
            return L;
        }
        double highest(){
            double M=rainfall[0];
            for(int i=0; i< this->cnt; i++){
                if(this->rainfall[i] > M){
                    most = i;
                    M = this->rainfall[i];
                }
            }
            return M;
        }
        bool storeValue(double rainfall){
            if (rainfall<0 || cnt>=30) return false;
            this->rainfall[this->cnt] = rainfall;
            cnt++;
            return true;
        }
        string convert(){
            stringstream ss;
            int monthnum;

            ss << this->monthYear.substr(4,2);
            ss >> monthnum;
            return this->month[monthnum] + ", " + this->monthYear.substr(0,4);
        }

        string convert(int cnt){
            stringstream ss;
            int monthnum, yearnum;
            string yearstr;

            ss << this->monthYear.substr(4,2);
            ss >> monthnum;
            ss.clear();

            ss << this->monthYear.substr(0,4);
            ss >> yearnum;
            ss.clear();

            yearnum += (monthnum + cnt)/12;
            monthnum = (monthnum + cnt)%12;

            ss << yearnum;
            ss >> yearstr;

            return this->month[monthnum] + ", " + yearstr;
        }
        void selectionSort(int array[], int size, int cnt[]){
```

```
PS 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.
13
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 > ...
        int start, minIndex, minValue, minCNT;
        for (start = 0; start < (size - 1); start++){
            minIndex = start;
            minValue = array[start];
            minCNT = cnt[start];
            for (int index = start + 1; index < size; index++){
                if (array[index] < minValue){
                    minValue = array[index];
                    minIndex = index;
                    minCNT   = cnt[index];
                }
            }
            array[minIndex] = array[start];
            array[start] = minValue;
            cnt[minIndex] = cnt[start];
            cnt[start] = minCNT;
        }
    }

    string displayReport(){
        string ret;
        ret += this->convert() + " -" + this->convert(cnt) + " Rain Report for " + this->city +" County\n";
        stringstream temp;
        string strtemp;
        temp<< total();
        temp>> strtemp;
        temp.clear();
        ret += "Total rainfall in this period: " + strtemp + " inches\n";
        temp<< average();
        temp>> strtemp;
        temp.clear();
        ret += "Average monthly rainfall: " + strtemp + " inches\n";
        temp<< lowest();
        temp>> strtemp;
        temp.clear();
        ret += "The least rain fell in " + this->convert(least) + " with " + strtemp + " inches\n";
        temp<< highest();
        temp>> strtemp;
        temp.clear();
        ret += "The most rain fell in " + this->convert(most) + " with " + strtemp + " inches\n";
        return ret;
    }
    string displayAscending(){
        string ret;
        ret += " Rainfall Report Display by ascending in " + this->city +" County : \n";
        ret += "=============================================================\n";
        int copyRainfall[this->cnt], cnt[this->cnt];
        for(int i=0; i<this->cnt; i++){
            copyRainfall[i] = this->rainfall[i];
            cnt[i] = i;
        }
        selectionSort(copyRainfall,this->cnt, cnt);
        for(int i=0; i<this->cnt; i++){
            stringstream temp;
            string strtemp;
            temp<< copyRainfall[i];
            temp>> strtemp;
            temp.clear();
            ret += convert(cnt[i])+"  Rainfall: "+ strtemp +"\n";
        }
        return ret;
    }
    string displayDescending(){
       string ret;
        ret += " Rainfall Report Display by descending in " + this->city +" County : \n";
        ret += "=============================================================\n";
        int copyRainfall[this->cnt], cnt[this->cnt];
        for(int i=0; i<this->cnt; i++){
            copyRainfall[i] = this->rainfall[i];
            cnt[i] = i;
        }
        selectionSort(copyRainfall,this->cnt, cnt);
        for(int i=this->cnt-1; i >= 0; i--){
            stringstream temp;
            string strtemp;
            temp<< copyRainfall[i];
            temp>> strtemp;
            temp.clear();
            ret += convert(cnt[i])+"  Rainfall: "+ strtemp +"\n";
        }
        return ret;
    }
    string displayTime(){
        string ret;
        ret += " Rainfall Report Display by time in " + this->city +" County : \n";
        ret += "=============================================================\n";
        for(int i=0; i<this->cnt; i++){
            stringstream temp;
            string strtemp;
            temp<< rainfall[i];
            temp>> strtemp;
```

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     }
```