

Purpose:

This assignment is the first assignment in this semester. I try to learn how to get more familiar with selection sort or bubble sort function. I tried to calculate the time of them to compare to each other. I can write how I should write the code for each algorithm and compare them together. I should figure out how to use an array or vector for coding. I think vectors are more appropriate because our text files are long. Moreover, learn the differences between them such as how long they need time to parse the data(sorted or unsorted). In the end, the user can ask to print the last 50 words from the sorted list.

Plan:

First of all I wrote two lines to greet and introduce the program and ask the user if she/he wants to continue or not. If she writes “y” it means the function continues and works but if she writes “n” it means to give up. I found the bubble sort code in the Microsoft team and also selected the

sort code in the textbook. I should clean the text file and read it and also push-back them in vector. (I chose vectors because I am not sure about the size of our file text. It is possible to do it by array but vector is better) Firstly, I am testing each function once it's done and once the program is done I'll test it with shortened text files to make sure it works in addition to the main text file. I should write a code to calculate the time of them so I can compare them together.

Input:

1. Users choose he/she want to continue or not (Yes or No).

Functions:

1. main function: I will read my three files in the main function and the end push-back them in the vector.
2. Void parse: and try to clean them
3. Void selectionSort: Find the smallest element and move it in index[0] and swap the value of index[0] to the last minimum index and continue.
4. Void bubbleSort: Compare every two indexes and swap the minimum of them on the left side.
5. Void print: this function prints the last 50 words.

Development process:

First part i wrote all libraries and function declarations

```
> baharehbahmani > Desktop > prj1 > main.cpp > ...
#include <iostream> // std::cin, std::cout
#include <fstream> // std::ifstream
#include <string> // std::string
#include <vector> // std::vector
#include <chrono> // for calculating the elapsed time
#include "sort.h" // importing sort functions declarations here, so compiler knows about these function.

/*
 * Compiler doesn't need to know about definition of the function
 * Linker links function calls to their definitions later
 */
```

//parse the text and clean it from everything instead of the alphabet.

```
//clean the text and change all letter to lower
void parse(std::string& str) {
    for (int i = str.size()-1; i >= 0; --i) {
        if (!std::isalpha(str[i])) {
            str.erase(str.begin() + i);
        }
        str[i] = std::tolower(str[i]);
    }
}
```

// read the text file , we have two parameter file(read the text) and words(create a vector)

```
20     }
21 }
22 //read the text
23 int read(std::fstream& file, std::vector<std::string>& words) {
24     int counter {};
25     while (file) {
26         std::string str;
27         file >> str;
28         parse(str);
29         if (str.size()) {
30             words.push_back(str);
31             counter++;
32         }
33     }
34
35     return counter;
36 }
```

Print the last 50 words of the text . In this function we have a parameter by reference.

```
}  
// print the last 50 words of text file  
void print(std::vector<std::string> &v) {  
    for (std::size_t i = v.size()-50; i < v.size(); i++) {  
        std::cout << v[i] << " ";  
    }  
    std::cout << '\n';  
}
```

//greeting and introduce the program

```
// greetings and introduction  
char greet() {  
    char ans;  
    std::cout << "\nHello\n" ;  
    std::cout << "Behjat bahmani, Lab1 - Sorting text using Selection sort and Bubble sort\n" ;  
    std::cout << "In this program we sort data using Selection sort and Bubble sort\n";  
    std::cout << "\ndo you want to continue: (y/n) ";  
    std::cin >> ans;  
  
    return ans;  
}
```

Main function reads the text file and also if our text doesn't exist we will get an error.

```
int main() {  
    std::ifstream input { "new.txt" };  
    //if our text file doesn't exist we will get this error  
    if (!input) {  
        std::cerr << "Error! Can't open the file.\n";  
        return 1;  
    }  
}
```

This is a while loop if that would be true and then call the function greet or If the answer be "n" .
"ans" is a character . After that, clean the text and push back the letters.

```
std::vector<std::string> sorted;
std::vector<std::string> words;
std::string str;

// greet and ask user to type y/n
while (true) {
    char ans = greet();
    if (ans == 'n') {
        return 0;
    }

    while (input >> str) {
        for (int i = 0, len = str.size(); i < len; i++) {
            if (ispunct(str[i])) {
                str.erase(i--, 1);
                len = str.size();
            }
        }
        words.push_back(str);
    }
}
```

I wrote the bubble sort function(the code of that is in sort.h) and also calculated the time of that.

```
std::cout << "\nstarting Bubble Sort...\n";
std::chrono::duration<double> elapsed_seconds;
auto start = std::chrono::system_clock::now();
bubbleSort(words);
auto end = std::chrono::system_clock::now();
elapsed_seconds = end - start;
std::cout << "Sorting unsorted vector took: " << elapsed_seconds.count() << " seconds" << '\n';
start = std::chrono::system_clock::now();
bubbleSort(words);
end = std::chrono::system_clock::now();
elapsed_seconds = end - start;
std::cout << "Sorting sorted vector took: " << elapsed_seconds.count() << " seconds" << "\n";
```

I wrote the selection sort function(the code of that is in sort.h) and also calculated the time of that.

```

    words = original;
    std::cout << "\nStarting Selection Sort...\n";
    start = std::chrono::system_clock::now();
    selectionSort(words);
    end = std::chrono::system_clock::now();
    elapsed_seconds = end - start;
    std::cout << "Sorting unsorted vector took: " << elapsed_seconds.count() << " seconds" << '\n';
    start = std::chrono::system_clock::now();
    selectionSort(words);
    end = std::chrono::system_clock::now();
    elapsed_seconds = end - start;
    std::cout << "Sorting sorted vector took: " << elapsed_seconds.count() << " seconds" << '\n';

    std::cout << "\nprinting last 50 words in sorted vector: " << '\n';
    print(words);
}
}

```

Pitfalls:

I didn't have so much of a problem this time in resubmitting. But last time I had some problems such as: I had some problems, first of all about cleaning the file from punctuation, it didn't work very well. I removed some spaces in the text file because the code didn't remove all of them, and also and also I had a problem with the length of the book because the code didn't read it very well. Moreover, about sorting in selection sort I had a probe how to change the arrays to vectors. And it didn't work for me.

Possible improvements:

I think it will be a good idea if I create a table to show each part separately so we can compare the time of them more easily. And also print the last 50 words in both sorted and unsorted text files.

Product:

```
MacBook-Pro:prj1 baharehbahmani$ g++ main.cpp sort.cpp -std=c++11 -o main
MacBook-Pro:prj1 baharehbahmani$
MacBook-Pro:prj1 baharehbahmani$ ./main

Hello
Behjat bahmani, Lab1 - Sorting text using Selection sort and Bubble sort
In this program we sort data using Selection sort and Bubble sort

do you want to continue: (y/n) y

starting Bubble Sort...
Sorting unsorted vector took: 16.4187 seconds
Sorting sorted vector took: 0.001321 seconds

Starting Selection Sort...
Sorting unsorted vector took: 6.61207 seconds
Sorting sorted vector took: 6.18082 seconds

printing last 50 words in sorted vector:
young young young younger your your your your your your your your your your your your your your your your your your your
your your your your your your youre youre youre youre youre youre yourself yourself yourself yourself youth youve youve youve

Hello
Behjat bahmani, Lab1 - Sorting text using Selection sort and Bubble sort
In this program we sort data using Selection sort and Bubble sort

do you want to continue: (y/n) ☐
```