

COSC 501

Lab 8

(30 points)Program 1: Sorting a string array

Write a C++ program that reads a list of 30 fruits from the file “fruits.txt”, inserts them into a string array, and sorts the array in alphabetical order. String objects can be compared using relational operators such as <, >, or ==. For example, (“abc” > “abd”) is false, but (“abc” < “abd”) is true.

Input: an input file name is “fruits.txt”.

Sample Output: Red colored texts are user inputs. Other texts are the output of the program.

Before Sorting:

Cherry, Honeydew, Cranberry, Lemon, Orange, Persimmon, Watermelon, Kiwifruit,
Lime, Pomegranate, Jujube, Pineapple, Durian, Plum, Banana, Coconut, Apple,
Tomato, Raisin, Mandarine, Blackberry, Raspberry, Peach, Mango, Melon, Grape,
Strawberry, Blueberry, Pear, Avocado

After Sorting:

Apple, Avocado, Banana, Blackberry, Blueberry, Cherry, Coconut, Cranberry,
Durian, Grape, Honeydew, Jujube, Kiwifruit, Lemon, Lime, Mandarine, Mango,
Melon, Orange, Peach, Pear, Persimmon, Pineapple, Plum, Pomegranate, Raisin,
Raspberry, Strawberry, Tomato, Watermelon

(30 points)Program 2: String and Vector

Write a C++ program that extracts words inside the parentheses from the text, and stores the words into a vector.

You will use this program:

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
using namespace std;
int main() {
    ifstream fin("Lab8_2data.txt");
    string text;
    vector<string> word;
    if(fin.fail()){
        cout << "Unable to read the file.";
        return -1;
    }
    getline(fin, text);
    // you need to implement from here
    fin.close();
    return 0;
}
```

What to use:

- `string.find(str, pos)` // returns the position of the first character of the first match, or -1 if no match.
// str: string to be searched for, pos: search from *pos*
- `string.substr(pos, len)` // returns a substring
// pos: position of the first character to be copied as a substring
// len: number of characters to include in the substring
- `vector.push_back(str)` // adds a new string element at the end of the vector
// str: string to be added
- `vector.at(n)` // returns the element at the specified position
// n: position of an element in the vector
- `vector.size()` // returns the number of elements in the vector

Sample Output: Red colored texts are user inputs. Other texts are the output of the program.

Text: Towson University (TU) is a Public, Non-Sectarian, University
established in 1866. The campus is located in Towson, Maryland, USA and hosts
21,960 (2012) students with an endowment of \$33.2 million (2010).

Extracted Word: TU 2012 2010

(40 points)Program 3: 2-dimension array

Write a program to assign passengers seats in an airplane. Assume a small airplane with seat numbering as follows:

```
1 A B C D
2 A B C D
3 A B C D
4 A B C D
5 A B C D
6 A B C D
7 A B C D
```

The program should display the seat pattern, with an X marking the seats already assigned. For example, after seats 1A, 2B, and 4C are taken, the display should look like this:

```
1 X B C D
2 A X C D
3 A B C D
4 A B X D
5 A B C D
6 A B C D
7 A B C D
```

After displaying the seats available, the program prompts for the seat desired, the user types in a seat, and then the display of available seats is updated. This continues until all seats are filled or until the user signals that the program should end. If the user types in a seat that is already assigned, the program should say that that seat is occupied and ask for another choice.

Submission:

You should submit your source files (.cpp). Please name your files to include the lab number and program number, e.g. Lab0Program1.cpp. Also create a word or pdf document for the answers to the lab questions and screenshot of running results.