PROG# 2 (3501)

In this program assignment, you are to implement an array of pointers to strings (or objects) as explained in Lafore ( pp. 474 -- ). Read the section on Pointers to pointers.

* Create a data file which contains the names from the PROG#1 data file. This time copy only the names with one name on each line.
* Read each name into a string [20].
* As you read the data from the file, construct an array of pointers to strings.

See Figure 10.17, Lafore p. 477).

* Sort the array of pointers in the ascending order of the names (dictionary order) using the bubble sort algorithm in Lafore. You are swapping pointers, not the names.
* Print out the sorted array of names. Upload a zip file containing the source file, solution space along with the output (do file I/O).

#include <iostream>

#include<iomanip>

#include<fstream>

#include <string>

using namespace std;

class student

{

private:

string name;

public:

student()

{

name = "";

}

void insert(string x)

{

name = x;

}

void print()

{

cout << name << endl;

}

string put()

{

return name;

}

friend void sort(student\* st[]);

};

void swap(char\* a, char\* b)

{

char t;

t = \*a;

\*a = \*b;

\*b = t;

}

void sort(student \*s[])

{

for (int i = 0; i < 24; i++)

{

for (int j = i; j <= 24; j++)

{

if (s[i]->name > s[j]->name)

swap(s[i], s[j]);

}

s[i]->print();

}

cout << "Sorted list: " << endl;

}

int main()

{

string x;

int c = 0;

ifstream in("C:/…………… txt");

student \*p1[25];

student s;

student st[25];

if (in.is\_open())

{

while (!in.eof())

{

in>>x;

st[c].insert(x);

p1[c] = &st[c];

c++;

}

cout << "MyName" << endl << endl;

sort(p1);

}

else

{

cout << "File cannot be opened.";

}

in.close();

// output file

ofstream out;

out.open("C:/………………… /output.txt");

if (out.is\_open())

{

out << ""<<endl;

out <<endl<< "sorted list: "<<endl;

for (int k = 0; k < 25; k++)

{

out << p1[k]->put() << endl;

}

}

else

{

cout << "File cannot be opened.";

}

out.close();

return 0;

}

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated