МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Західноукраінський національний університет

Кафедра Інформаційно-Обчислювальних Систем і Управління

Групи КНШІ-11

**Лабораторна робота №5**

Виконала

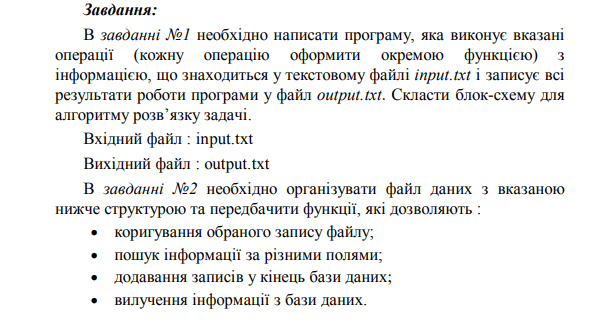
Неділь В.Р.

Перевірив

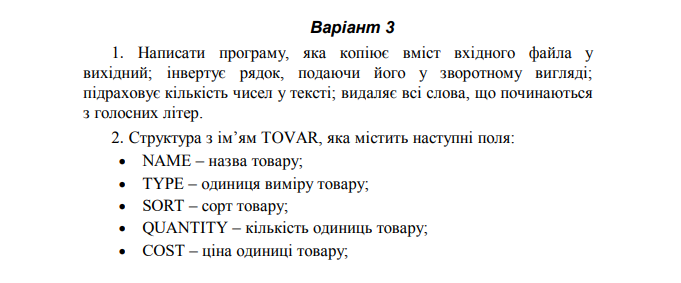
Цапів Я.А.

Тернопіль

2024 рік



***Мета:*** навчитися складати програми для роботи з файлами заданої структури та виконання обробки текстової інформації.



*Завдання 1.*

*Код:*

#include <iostream>

#include <string>

#include <fstream>

using namespace std;

string invertText(string);

bool isVowel(char);

bool isDigit(char);

bool isNumber(string);

int numCounter(string);

string deleteWords(string);

int main()

{

ifstream input("input.txt");

string text = "";

while (input)

{

string row = "";

getline(input, row);

text += row;

}

string invertedText = invertText(text);

int numberCount = numCounter(text);

string newText = deleteWords(text);

ofstream outfile("output.txt");

outfile << "Inverted text: " << invertedText << endl;

outfile << "Number of numbers: " << numberCount << endl;

outfile << "Text without words which starts at vowel letter: " << newText << endl;

}

string invertText(string text)

{

string result = "";

string word = "";

for (char character : text)

{

if (character == ' ')

{

result = word + " " + result;

word = "";

}

else {

word += character;

}

}

result = word + " " + result; // for last word

return result;

}

bool isVowel(char character)

{

char vowels[6]{ 'a', 'e', 'o', 'i', 'u', 'y' };

for (int i = 0; i < 6; i++)

if (tolower(character) == vowels[i]) return true;

return false;

}

bool isDigit(char character)

{

for (int digit = 0; digit < 10; digit++)

if ('0' + digit == character) return true;

return false;

}

bool isNumber(string word)

{

bool isNum = true;

for (char symbol : word)

{

if (!isDigit(symbol)) isNum = false;

}

return isNum;

}

int numCounter(string text)

{

int numberCount = 0;

string word = "";

for (char character : text)

{

if (character == ' ')

{

if (isNumber(word)) numberCount++;

word = "";

}

else {

word += character;

}

}

if (isNumber(word)) numberCount++; // for last word

return numberCount;

}

string deleteWords(string text)

{

string result = "";

string word = "";

for (char character : text)

{

if (character == ' ')

{

if (!isVowel(word[0])) result += word + " ";

word = "";

}

else {

word += character;

}

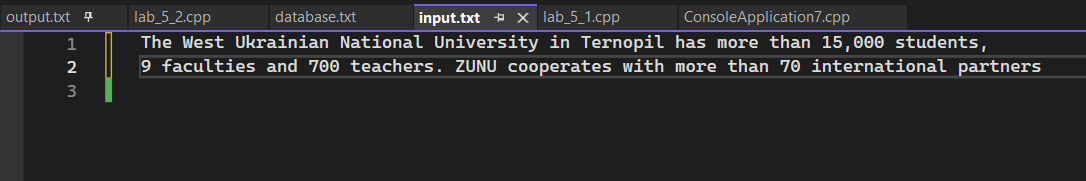
}

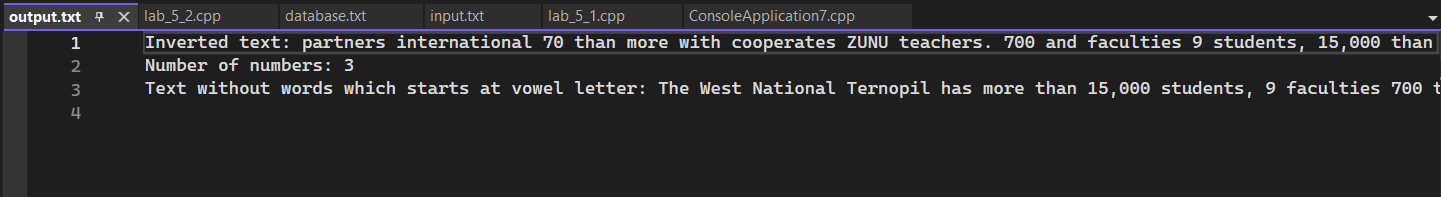
if (!isVowel(word[0])) result += word + " "; // for last word

return result;

}

*Контрольний приклад:*

**

**

*Завдання 2.*

*Код:*

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

#include <cstdlib>

using namespace std;

struct Tovar

{

string name;

string type;

string sort;

int quantity;

string cost;

};

void displayBase(vector<Tovar>&);

void addRecord(vector<Tovar>&);

void deleteRecord(vector<Tovar>&);

void editRecord(vector<Tovar>&);

void searchRecord(vector<Tovar>&);

int main()

{

ifstream databaseFile("database.txt");

vector<Tovar> database;

while (databaseFile)

{

Tovar record;

string stringRecord;

getline(databaseFile, stringRecord);

if (stringRecord == "") break;

string key = "";

string value = "";

string currentText = "";

for (char ch : stringRecord)

{

if (ch == ';') value = currentText;

if (key != "" && value != "")

{

if (key == "name") record.name = value;

else if (key == "type") record.type = value;

else if (key == "sort") record.sort = value;

else if (key == "quantity") record.quantity = stoi(value);

else if (key == "cost") record.cost = value;

key = "";

value = "";

}

if (ch == ';') break;

if (ch == ':')

{

key = currentText;

currentText = "";

}

else if (ch == ',')

{

value = currentText;

currentText = "";

}

else

{

currentText += ch;

}

}

database.push\_back(record);

}

while (true)

{

int operation;

cout << "Enter a command" << endl;

cout << "(0-exit | 1-add record | 2-delete record | 3-edit record | 4-search record | 5-display database)" << endl;

cin >> operation;

switch (operation)

{

case 0: return 0;

case 1:

addRecord(database);

break;

case 2:

deleteRecord(database);

break;

case 3:

editRecord(database);

break;

case 4:

searchRecord(database);

break;

case 5:

displayBase(database);

break;

default: return 0;

}

}

return 0;

}

void displayBase(vector<Tovar>& database)

{

for (int i = 0; i < database.size(); i++)

{

Tovar record = database[i];

cout << "=========" << i + 1 << "=========" << endl;

cout << "Name | " << record.name << endl;

cout << "Type | " << record.type << endl;

cout << "Sort | " << record.sort << endl;

cout << "Quantity | " << record.quantity << endl;

cout << "Cost | " << record.cost << endl;

if (i == database.size() - 1) cout << endl;

}

}

void addRecord(vector<Tovar>& database)

{

Tovar newRecord;

cout << "Enter name of product: ";

cin >> newRecord.name;

cout << "Enter unit of measurement of product: ";

cin >> newRecord.type;

cout << "Enter sort of product: ";

cin >> newRecord.sort;

cout << "Enter quantity of products: ";

cin >> newRecord.quantity;

cout << "Enter cost of product: ";

cin >> newRecord.cost;

database.push\_back(newRecord);

ofstream databaseFile("database.txt", ios::app);

databaseFile << "name:" << newRecord.name << ",";

databaseFile << "type:" << newRecord.type << ",";

databaseFile << "sort:" << newRecord.sort << ",";

databaseFile << "quantity:" << newRecord.quantity << ",";

databaseFile << "cost:" << newRecord.cost << ";" << endl;

}

void deleteRecord(vector<Tovar>& database)

{

int index;

cout << "Enter which record to delete: ";

cin >> index;

vector<Tovar> newDatabase;

for (int i = 0; i < database.size(); i++)

{

if (index - 1 == i) continue;

newDatabase.push\_back(database[i]);

}

database = newDatabase;

ofstream databaseFile("database.txt", ios::trunc);

for (Tovar record : database)

{

databaseFile << "name:" << record.name << ",";

databaseFile << "type:" << record.type << ",";

databaseFile << "sort:" << record.sort << ",";

databaseFile << "quantity:" << record.quantity << ",";

databaseFile << "cost:" << record.cost << ";" << endl;

}

}

void editRecord(vector<Tovar>& database)

{

int index, choice;

cout << "Enter which record to edit: ";

cin >> index;

cout << "What will edit? (0 - Exit from Edit Mode | 1 - name | 2 - type | 3 - sort | 4 - quantity | 5 - cost): ";

cin >> choice;

switch (choice)

{

case 0: return;

case 1:

cout << "Enter new name: ";

cin >> database[index - 1].name;

break;

case 2:

cout << "Enter new type: ";

cin >> database[index - 1].type;

break;

case 3:

cout << "Enter new sort: ";

cin >> database[index - 1].sort;

break;

case 4:

cout << "Enter new quantity: ";

cin >> database[index - 1].quantity;

break;

case 5:

cout << "Enter new cost: ";

cin >> database[index - 1].cost;

break;

default: return;

}

ofstream databaseFile("database.txt", ios::trunc);

for (Tovar record : database)

{

databaseFile << "name:" << record.name << ",";

databaseFile << "type:" << record.type << ",";

databaseFile << "sort:" << record.sort << ",";

databaseFile << "quantity:" << record.quantity << ",";

databaseFile << "cost:" << record.cost << ";" << endl;

}

}

void searchRecord(vector<Tovar>& database)

{

string str\_searchedValue;

int int\_searchedValue;

int choice;

cout << "Enter what seacrh (1 - name | 2 - type | 3 - sort | 4 - quantity | 5 - cost): ";

cin >> choice;

cout << "Enter value to seacrh for: ";

if (choice == 4) cin >> int\_searchedValue;

else cin >> str\_searchedValue;

vector<Tovar> searchedRecords;

switch (choice)

{

case 1:

for (Tovar record : database)

if (record.name == str\_searchedValue) searchedRecords.push\_back(record);

break;

case 2:

for (Tovar record : database)

if (record.type == str\_searchedValue) searchedRecords.push\_back(record);

break;

case 3:

for (Tovar record : database)

if (record.sort == str\_searchedValue) searchedRecords.push\_back(record);

break;

case 4:

for (Tovar record : database)

if (record.quantity == int\_searchedValue) searchedRecords.push\_back(record);

break;

case 5:

for (Tovar record : database)

if (record.cost == str\_searchedValue) searchedRecords.push\_back(record);

break;

default: return;

}

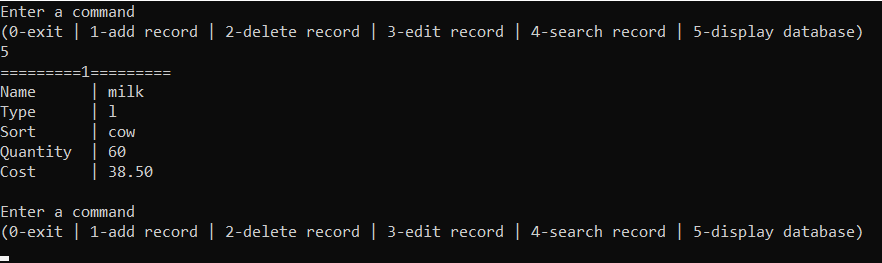
cout << "Search.." << endl;

if (searchedRecords.size() == 0) cout << "Records not found..." << endl;

displayBase(searchedRecords);

}

*Контрольний приклад:*

**

***Висновок.*** Отже, ми навчилися складати програми для роботи з файлами заданої структури та виконання обробки текстової інформації.