

Міністерство освіти і науки України

КПІ ім. Ігоря Сікорського

Кафедра ІІІ

ЗВІТ

з виконання лабораторної роботи № 2

з кредитного модуля

“Основи програмування-2. Методології програмування”

Варіант № 1

Виконав:

студент 1-го курсу

гр. ІІІ-22 ФІОТ

Андрєєва Уляна Андріївна

Київ 2023

ПОСТАНОВКА ЗАДАЧІ

Створити файл з переліком продуктів у магазині: найменування, дата випуску, кінцевий термін придатності. Створити два нових файли, в один з яких помістити перелік продуктів, що швидко псуються (термін зберігання яких не більше 5-ти діб). в другий - продуктів тривалого зберігання.

Видалити із створених файлів продукти, термін яких закінчився.

main.cpp

```
#include "Functions.h"

int main() {
    time_t desirableDateUnix = time(0);

    int choice;
    cout << "Enter 1, if you want to use appending, otherwise enter any number here: ";
    cin >> choice;

    ofstream File1;
    ofstream File2;
    ofstream File3;

    if (choice == 1) {
        File1.open("productList.bin", ios::app | ios::binary);
        File2.open("perishableProducts.bin", ios::app | ios::binary);
        File3.open("nonperishableProducts.bin", ios::app | ios::binary);
    }
    else {
        File1.open("productList.bin", ios::binary);
        File2.open("perishableProducts.bin", ios::binary);
        File3.open("nonperishableProducts.bin", ios::binary);
    }

    string text = captureText();
    int linesCount = count(text.begin(), text.end(), '\n');
    string linesArray[linesCount];
    DivideLines(text, linesCount, linesArray);

    Product products[linesCount];
    GetProducts( linesArray, linesCount, products);

    RecordProducts( linesCount , File2,File3, File1, products);

    File1.close();
    File2.close();
    File3.close();

    RemoveExpiredProducts(desirableDateUnix, "productList.bin");
    RemoveExpiredProducts(desirableDateUnix, "perishableProducts.bin");
    RemoveExpiredProducts(desirableDateUnix, "nonperishableProducts.bin");
}
```

Functions.cpp

```
#include <string>
#include <sstream>
#include <fstream>
#include <iostream>
#include "Functions.h"

Product ConvertStringToProduct(string line){
    int comaPosition = line.find(',');
    string name = line.substr(0, comaPosition - 0);
    int comaPosition2 = line.rfind(',');
    string productionDateString = line.substr(comaPosition + 1,
comaPosition2 - comaPosition - 1);

    struct tm tm{};
    istringstream ss(productionDateString);
    ss >> get_time(&tm, "%d.%m.%y");

    time_t productionDate = mktime(&tm);

    int newLinePosition2 = line.rfind('\n');
    string expirationDateString = line.substr(comaPosition2 + 1,
newLinePosition2 - comaPosition2);
    struct tm tm2{};
    istringstream ss2(expirationDateString);
    ss2 >> get_time(&tm2, "%d.%m.%y");
    time_t expirationDate = mktime(&tm2);

    struct Product product{name, productionDate, expirationDate};

    return product;
}

void RemoveExpiredProducts(time_t desirableDateUnix, const char *fileName){
    string line;
    ifstream File(fileName, ios::binary);
    ofstream FileTemp("temp.bin", ios::binary);
    while(getline(File, line)){
        Product product = ConvertStringToProduct(line);

        if(desirableDateUnix<=product.expirationDate){
            FileTemp << line << endl;
        }
    }

    File.close();
    FileTemp.close();
    remove(fileName);
    rename("temp.bin", fileName);
}

void RecordProducts(int linesCount , ofstream &File2, ofstream &File3,
ofstream &File1, Product products[]) {
    for (int i = 0; i < linesCount; ++i) {
        RecordAll(File1, products, i);
    }
}
```

```

        //432000 - 5 days in seconds
        if (products[i].expirationDate - products[i].productionDate <
432000) {
            string str = ConvertProductToString(products[i]);
            cout << "Product: " << str << " is perishable " << endl;
            RecordPerishable(File2, products, i);

            // Write the string to the file
        } else {
            string str = ConvertProductToString(products[i]);
            cout << "Product: " << str << " is nonperishable " << endl;
            RecordNonperishable(File3, products, i);
        }
    }
}

void RecordAll(ofstream &File1, Product products[], int i) {
    string str = ConvertProductToString(products[i]);
    int bufferSize = sizeof(str[0]) * str.size();
    File1.write((char *) &str[0], bufferSize);
}

string ConvertProductToString(Product product){
    struct tm *expiration_date;
    char buffer[80];
    expiration_date = localtime(&product.expirationDate);
    strftime(buffer, sizeof(buffer), "%d.%m.%Y", expiration_date);

    struct tm *production_date;
    char buffer2[80];
    production_date = localtime(&product.productionDate);
    strftime(buffer2, sizeof(buffer2), "%d.%m.%Y", production_date);
    string str = product.name + ',' + (buffer2) + ',' + (buffer) + '\n';

    return str;
}

void RecordNonperishable(ofstream &File3, Product products[], int i) {
    string productStr = ConvertProductToString(products[i]);

    int bufferSize = sizeof(productStr[0]) * productStr.size();
    File3.write((char *) &productStr[0], bufferSize);
}

void RecordPerishable(ofstream &File2, Product products[], int i) {
    string str = ConvertProductToString(products[i]);

    int bufferSize = sizeof(str[0]) * str.size(); // Define the buffer
size for the string
    File2.write((char*)&str[0], bufferSize);
}

void GetProducts(string linesArray[], int linesCount, Product products[]) {
    for (int i = 0; i < linesCount; ++i) {
        Product product = ConvertStringToProduct(linesArray[i]);
        products[i] = product;
    }
}

void DivideLines(const string &text, int linesCount, string *linesArray) {
    int newLinePositionCounter = 0;

```

```

        for (int i = 0; i < linesCount; ++i) {
            int newLinePosition = text.find('\n', newLinePositionCounter+1);
            string NewText;

            if(i == 0){
                NewText = text.substr(newLinePositionCounter, newLinePosition-
newLinePositionCounter);
            }
            else{
                NewText = text.substr(newLinePositionCounter+1,
newLinePosition-newLinePositionCounter-1);
            }

            newLinePositionCounter = text.find('\n', newLinePositionCounter+1);
            linesArray[i] = NewText;
        }
    }

string captureText(){
    cout << "Enter your text here (an example is -
yogurt,21.03.2023,27.03.2023);\nPress Enter to go to the next line;\nPress
'-' + Enter on the new line to end the writing;\n";
    string text;
    string line;
    // ascii код символу "-"
    int asciiCode = 45;
    // поки перший символ строки не "-"
    while(int(line[0]) != asciiCode){
        getline(cin, line);
        text += line + '\n';
    }
    // відкидаємо 3 зайвих символи (2 зайвих \n і '-')
    text = text.substr(0, text.size()-2);

    cout << "Entered products: " << text << endl;

    return text;
}

```

Functions.h

```

#ifndef LABWORK_2__FUNCTIONS_H
#define LABWORK_2__FUNCTIONS_H

#include <ctime>
#include <iostream>
#include <fstream>
#include <sstream>
#include <string>
#include "Structure.h"
using namespace std;

Product ConvertStringToProduct(string line);

void RemoveExpiredProducts(time_t desirableDateUnix, const char *fileName);

void RecordProducts(int linesCount , ofstream &File2, ofstream &File3,
ofstream &File1, Product products[]);
void RecordAll(ofstream &File1, Product products[], int i);

```

```

string ConvertProductToString(Product product);
void RecordNonperishable(ofstream &File3, Product products[], int i);

void RecordPerishable(ofstream &File2, Product products[], int i);

void GetProducts(string linesArray[], int linesCount, Product products[]);
void DivideLines(const string &text, int linesCount, string *linesArray);
string captureText();

#endif //LABWORK_2__FUNCTIONS_H

```

Structure.h

```

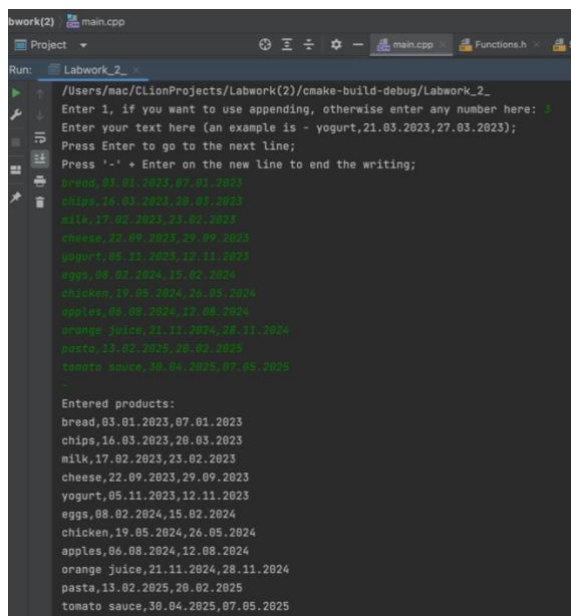
#include <ctime>
#include <string>
using namespace std;

struct Product {
    string name;
    time_t productionDate;
    time_t expirationDate;
};

```

РЕЗУЛЬТАТИ ТЕСТУВАННЯ

Результат консолі



```

Labwork(2)  main.cpp
Project
Run: Labwork_2_
/Users/mac/CLionProjects/Labwork(2)/cmake-build-debug/Labwork_2_
Enter 1, if you want to use appending, otherwise enter any number here: 1
Enter your text here (an example is - yogurt,21.03.2023,27.03.2023);
Press Enter to go to the next line;
Press '-' + Enter on the new line to end the writing;
bread,03.01.2023,07.01.2023
chips,16.03.2023,20.03.2023
milk,17.02.2023,23.02.2023
cheese,22.09.2023,29.09.2023
yogurt,05.11.2023,12.11.2023
eggs,08.02.2024,15.02.2024
chicken,19.05.2024,26.05.2024
apples,06.08.2024,12.08.2024
orange juice,21.11.2024,28.11.2024
pasta,13.02.2025,20.02.2025
tomato sauce,30.04.2025,07.05.2025
-
Entered products:
bread,03.01.2023,07.01.2023
chips,16.03.2023,20.03.2023
milk,17.02.2023,23.02.2023
cheese,22.09.2023,29.09.2023
yogurt,05.11.2023,12.11.2023
eggs,08.02.2024,15.02.2024
chicken,19.05.2024,26.05.2024
apples,06.08.2024,12.08.2024
orange juice,21.11.2024,28.11.2024
pasta,13.02.2025,20.02.2025
tomato sauce,30.04.2025,07.05.2025

```

```

Product:
bread,03.01.2023,07.01.2023
is perishable
Product: chips,16.03.2023,20.03.2023
is perishable
Product: milk,17.02.2023,23.02.2023
is nonperishable
Product: cheese,22.09.2023,29.09.2023
is nonperishable
Product: yogurt,05.11.2023,12.11.2023
is nonperishable
Product: eggs,08.02.2024,15.02.2024
is nonperishable
Product: chicken,19.05.2024,26.05.2024
is nonperishable
Product: apples,06.08.2024,12.08.2024
is nonperishable
Product: orange juice,21.11.2024,28.11.2024
is nonperishable
Product: pasta,13.02.2025,20.02.2025
is nonperishable
Product: tomato sauce,30.04.2025,07.05.2025
is nonperishable
Product: ,01.01.1970,01.01.1970
is perishable

Process finished with exit code 0

```

Результат вмісту файлу

- Nonperishable products

```

1 cheese,22.09.2023,29.09.2023
2 yogurt,05.11.2023,12.11.2023
3 eggs,08.02.2024,15.02.2024
4 chicken,19.05.2024,26.05.2024
5 apples,06.08.2024,12.08.2024
6 orange juice,21.11.2024,28.11.2024
7 pasta,13.02.2025,20.02.2025
8 tomato sauce,30.04.2025,07.05.2025
9

```

In a Binary mode

Hex	Decimal	Character
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F		
00 63 68 65 65 73 65 2C 32 32 2E 30 39 2E 32 30 32	99	cheese,22.09.2023
10 33 2C 32 39 2E 30 39 2E 32 30 32 33 79 6F 67	25448	3,29.09.2023 yog
20 75 72 74 2C 30 35 2E 31 31 2E 32 30 32 33 2C 31	1667786085	urt,05.11.2023,1
30 32 2E 31 31 2E 32 30 32 33 65 67 67 73 2C 30	7163086693734886450	2.11.2023 eggs,0
40 38 2E 30 32 2E 32 30 32 34 2C 31 35 2E 30 32 2E	4.286951E21	8.02.2024,15.02.
50 32 30 32 34 63 68 69 63 68 65 6E 2C 31 39 2E	7.365590916963406E170	2024 chicken,19.
60 38 35 2E 32 30 32 34 2C 32 36 2E 30 35 2E 32 30	c	05.2024,26.05.20
70 32 34 61 70 70 6C 65 73 2C 30 36 2E 30 38 2E		24 apples,06.08.
80 32 30 32 34 2C 31 32 2E 30 38 2E 32 30 32 34		2024,12.08.2024
90 6F 72 61 6E 67 65 20 6A 75 69 63 65 2C 32 31 2E		orange juice,21.
A0 31 31 2E 32 30 32 34 2C 32 38 2E 31 31 2E 32 30		11.2024,28.11.20
B0 32 34 70 61 73 74 61 2C 31 33 2E 30 32 2E 32		24 pasta,13.02.2
C0 30 32 35 2C 32 30 2E 30 32 2E 32 30 32 35		74 025,20.02.2025 t
D0 6F 6D 61 74 6F 20 73 61 75 63 65 2C 33 30 2E		0 tomato sauce,30.0
E0 34 2E 32 30 32 35 2C 30 37 2E 30 35 2E 32 30 32		4.2025,07.05.202
F0 35		5

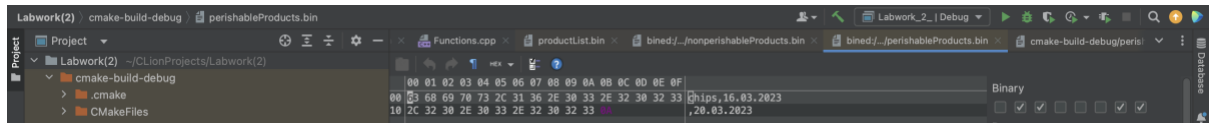
- perishable products

```

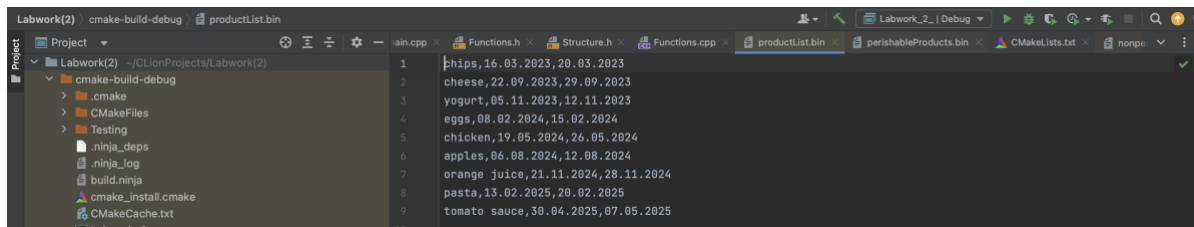
1 chips,16.03.2023,20.03.2023
2

```

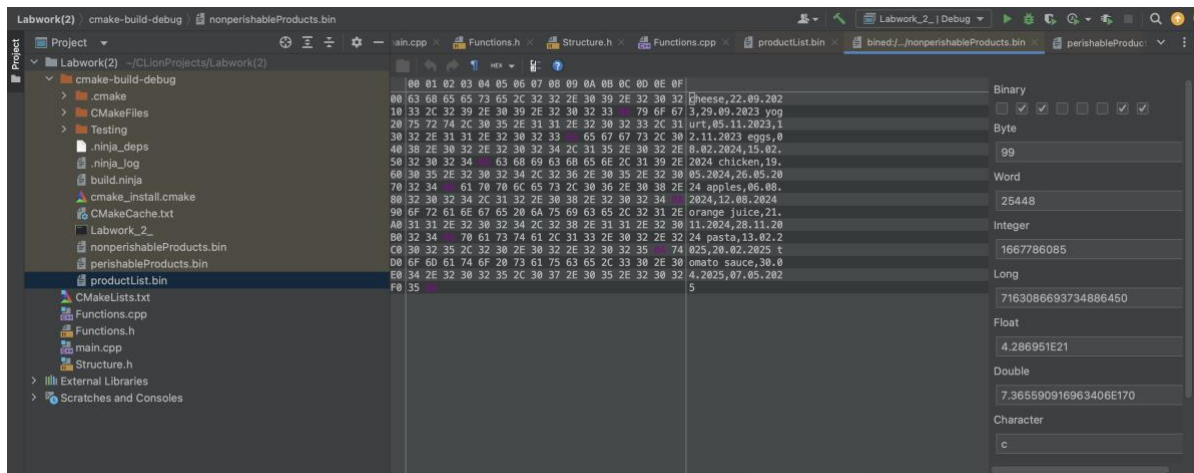
In a Binary mode



- productList



In a Binary mode



Лінк на репозиторій у GitHub:

<https://github.com/Uliana200407/CppProjects-.git>