

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Механіко-технологічний факультет

ЗВІТ
ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 10
з навчальної дисципліни
“Базові методології та технології програмування”

Реалізація програмних модулів оброблення даних складових типів з файловим
введенням/виведенням

ЗАВДАННЯ ВИДАВ
доцент кафедри кібербезпеки
та програмного забезпечення
Доренський О. П.
<https://github.com/odorenskyi/>

ВИКОНАВ
студент академічної групи КІ-22-1
Гой Д. О.

ПЕРЕВІРИВ
ст. викладач кафедри кібербезпеки
та програмного забезпечення
Собінов О. Г.

Мета: набуття ґрунтовних вмінь і практичних навичок реалізації у Code::Blocks IDE мовою програмування C++ програмних модулів створення й оброблення даних типів масив, структура, об'єднання, множина, перелік, перетворення типів даних, використання файлових потоків та функцій стандартних бібліотек для оброблення символічної інформації

Варіант 6

Завдання 1.

У вихідний текстовий файл записати:

- авторську інформацію: ім'я й прізвище розробника модуля, установа/організація, місто, країна, рік розробки;
- кількість знаків «!» та «?» (кожного окремо) у реченні із вхідного файлу;
- речення із вхідного файлу, у якому відсутні всі голосні літери.

Завдання 2.

У вихідний текстовий файл дописати:

- транслітерований текст з цього файлу відповідно до Постанови Кабінету Міністрів України “Про впорядкування транслітерації українського алфавіту латиницею” від 27 січня 2010 року № 55 (див. на звороті);
- дату й час дозапису інформації (транслітерованого тексту).

Завдання 3.

Вхідні дані – числові значення x , y , z та натуральне число b . У вихідний текстовий файл дописати:

- результати виконання функцій із заголовкового файлу `Modules/Прізвище.h` `s_calculation` з аргументами x , y , z ;
- число b у двійковому коді.

Лістинг `ModulesHoi.cpp`:

```
#include <iostream>
#include <cmath>
#include <string>
#include <fstream>
#include <windows.h>
#include <ctime>

using namespace std;
```

```

float s_calculation(float values[])
{
    if(values[0] - values[1] != 0 && (2 * pow(values[1], 2)) != 0 && (values[0] +
(values[2] /(2 * pow(values[1], 2)))) != 0 && (M_PI * pow(values[0], 2)) / (values[0] +
(values[2] /(2 * pow(values[1], 2)))) >= 0))
    {
        float S = log(values[0] - values[1]) + sqrt((M_PI * pow(values[0], 2)) /
(values[0] + (values[2] /(2 * pow(values[1], 2)))));
        return S;
    }
}

string Beaufort_score(float speed)
{
    if(speed < 0.3)
        return "Бал Бофорта: 0. Відсутність вітру. Дим піднімається прямовисно. Листя
дерев нерухомі.";
    if(speed >= 0.3 && speed <= 1.5)
        return "Бал Бофорта: 1. \"Дим пливе\". Флюгер не обертається.";
    if(speed >= 1.6 && speed <= 3.3)
        return "Бал Бофорта: 2. Рух повітря відчувається обличчям. Шелестить листя.
Флюгер обертається.";
    if(speed >= 3.4 && speed <= 5.4)
        return "Бал Бофорта: 3. Тріпоче листя, хитаються дрібні гілки. Майорять
прапори.";
    if(speed >= 5.5 && speed <= 7.9)
        return "Бал Бофорта: 4. Хитаються тонкі гілки дерев. Вітер піднімає пил та
шматки паперу.";
    if(speed >= 8.0 && speed <= 10.7)
        return "Бал Бофорта: 5. Хитаються великі гілки. На воді з'являються хвилі.";
    if(speed >= 10.8 && speed <= 13.8)
        return "Бал Бофорта: 6. Хитаються великі гілки.";
    if(speed >= 13.9 && speed <= 17.1)
        return "Бал Бофорта: 7. Хитаються невеликі стовбури дерев. На морі здіймаються
хвилі, піняться.";
    if(speed >= 17.2 && speed <= 20.7)
        return "Бал Бофорта: 8. Ламаються гілки дерев. Важко йти проти вітру.";
    if(speed >= 20.8 && speed <= 24.4)
        return "Бал Бофорта: 9. Невеликі руйнування. Зриває покрівлі, руйнує димарі.";
    if(speed >= 24.5 && speed <= 28.4)

```

```

        return "Бал Бофорта: 10. Значні руйнування. Дерева вириваються з корінням.";
    if(speed >= 28.5 && speed <= 32.6)
        return "Бал Бофорта: 11. Великі руйнування.";
    if(speed >= 32.7)
        return "Бал Бофорта: 12. Призводить до спустошень.";
}

string average_temperature(float temperatures[], int days)
{
    string result = "";

    float Celsius = 0;
    float Fahrenheit;

    for(int i = 0; i < days; i++)
        Celsius += temperatures[i];

    Celsius /= days;
    Fahrenheit = (1.8 * Celsius) + 32;

    result = "Середньомісячна температура у Цельсіях: " + to_string(Celsius) + ".
    Середньомісячна температура у Фаренгейтах: " + to_string(Fahrenheit);
    return result;
}

string amount_binary_digits(int N)
{
    string result = "";

    if(N < 0 || N > 10008000)
    {
        result = "Помилка. Некоректне значення.";
        return result;
    }
    else
    {
        int bit = (N >> 13) & 1;
        int amount = 0;

        if(bit == 1)
        {

```

```

        while((N & 1) == 0)
        {
            amount++;
            N >>= 1;
        }

        result = "Кількість двійкових нулів: " + to_string(amount);
        return result;
    }
    else
    {
        while (N != 0)
        {
            if (N & 1)
                amount++;
            N >>= 1;
        }

        result = "Кількість двійкових одиниць: " + to_string(amount);
        return result;
    }
}
}
}

```

```

void task_10_1(string file)
{
    SetConsoleCP(1251); SetConsoleOutputCP(1251);

    try
    {
        ifstream file_input(file);

        if (!file_input.is_open())
            throw "Помилка: не вдалося відкрити вхідний файл.";

        ofstream file_output("Ofile.txt");

        if (!file_output.is_open())
            throw "Помилка: не вдалося відкрити вихідний файл.";
    }
}

```

```
file_output << "Розробник ПЗ: Гой Дмитро Олександрович; ЗВО:  
Центральноукраїнський національний технічний університет";
```

```
file_output << " Країна: Україна; Місто: Знам'янка; Рік: 2023.\n";
```

```
string text = "";
```

```
int amount_questions = 0, amount_exclamations = 0;
```

```
bool sentences_with_vowels = false;
```

```
getline(file_input, text);
```

```
for(int i = 0; i < sizeof(text); i++)
```

```
{
```

```
    switch(text[i])
```

```
    {
```

```
        case 'a':
```

```
        case 'A':
```

```
        case 'y':
```

```
        case 'Y':
```

```
        case 'e':
```

```
        case 'E':
```

```
        case 'й':
```

```
        case 'Й':
```

```
        case 'o':
```

```
        case 'O':
```

```
        case 'є':
```

```
        case 'Є':
```

```
        case 'и':
```

```
        case 'И':
```

```
        case 'я':
```

```
        case 'Я':
```

```
        case 'ю':
```

```
        case 'Ю':
```

```
            sentences_with_vowels = true;
```

```
            break;
```

```
        case '\\?':
```

```
            amount_questions++;
```

```
            break;
```

```
        case '\\!':
```

```
            amount_exclamations++;
```

```
            break;
```

```
    }
```

```
}
```

```

file_output << "\nКількість \"?\": " << amount_questions << endl;
file_output << "Кількість \"!\": " << amount_exclamations << "\n\n";

if (!sentences_with_vowels)
    file_output << "Речення в якому немає голосних:\n\t\t" << text << endl;

file_output.close();
file_input.close();
}
catch(const char *error)
{
    cout << error << endl;
}
}

void task_10_2(string file)
{
    try
    {
        fstream file_input(file, ios_base::in | ios_base::app);

        if (!file_input.is_open())
            throw "Помилка: не вдалося відкрити файл.";

        string text = "";
        int i;

        getline(file_input, text);

        file_input.seekg(0, ios::end);

        file_input << endl;

        for (i = 0; i < size(text); i++)
        {
            switch(text[i])
            {
                case 'Й':
                    if(text[i-1] == ' ' || i == 0)
                        file_input << "Y";
            }
        }
    }
}

```

```

        break;
    case 'й':
        if(text[i-1] == ' ' || i == 0)
            file_input << "y";
        else
        {
            file_input << "i";
        }
        break;
    case 'Ц':
        file_input << "Ts";
        break;
    case 'ц':
        file_input << "ts";
        break;
    case 'Y':
        file_input << "U";
        break;
    case 'y':
        file_input << "u";
        break;
    case 'K':
        file_input << "K";
        break;
    case 'к':
        file_input << "k";
        break;
    case 'H':
        file_input << "N";
        break;
    case 'h':
        file_input << "n";
        break;
    case 'Г':
        file_input << "H";
        break;
    case 'г':
        file_input << "h";
        break;
    case 'Г':
        file_input << "G";

```



```

        break;
    case 'r':
        file_input << "g";
        break;
    case 'Ш':
        file_input << "Sh";
        break;
    case 'ш':
        file_input << "sh";
        break;
    case 'Щ':
        file_input << "Shch";
        break;
    case 'щ':
        file_input << "shch";
        break;
    case 'З':
        if(text[i+1] == 'r')
            file_input << "Zgh";
        else
        {
            file_input << "Z";
        }
        break;
    case 'з':
        if(text[i+1] == 'r')
            file_input << "zgh";
        else
        {
            file_input << "z";
        }
        break;
    case 'X':
        file_input << "Kh";
        break;
    case 'x':
        file_input << "kh";
        break;
    case 'İ':
        if(text[i-1] == ' ' || i == 0)
            file_input << "Yi";

```

```

        break;
case 'i':
    if(text[i-1] == ' ' || i == 0)
        file_input << "yi";
    else
    {
        file_input << "i";
    }
    break;
case 'Ф':
    file_input << "F";
    break;
case 'ф':
    file_input << "f";
    break;
case 'В':
    file_input << "V";
    break;
case 'в':
    file_input << "v";
    break;
case 'П':
    file_input << "P";
    break;
case 'п':
    file_input << "p";
    break;
case 'Р':
    file_input << "R";
    break;
case 'р':
    file_input << "r";
    break;
case 'Л':
    file_input << "L";
    break;
case 'л':
    file_input << "l";
    break;
case 'Д':
    file_input << "D";

```

```

        break;
case 'д':
    file_input << "d";
    break;
case 'Ж':
    file_input << "Zh";
    break;
case 'ж':
    file_input << "zh";
    break;
case 'Є':
    if(text[i-1] == ' ' || i == 0)
        file_input << "Ye";
    break;
case 'є':
    if(text[i-1] == ' ' || i == 0)
        file_input << "ye";
    else
    {
        file_input << "ie";
    }
    break;
case 'Я':
    if(text[i-1] == ' ' || i == 0)
        file_input << "Ya";
    break;
case 'я':
    if(text[i-1] == ' ' || i == 0)
        file_input << "ya";
    else
    {
        file_input << "ia";
    }
    break;
case 'Ч':
    file_input << "Ch";
    break;
case 'ч':
    file_input << "ch";
    break;
case 'C':

```

```

        file_input << "S";
        break;
case 'c':
    file_input << "s";
    break;
case 'м':
    file_input << "m";
    break;
case 'И':
    file_input << "Y";
    break;
case 'и':
    file_input << "y";
    break;
case 'т':
    file_input << "t";
    break;
case 'Б':
    file_input << "B";
    break;
case '6':
    file_input << "b";
    break;
case 'Ю':
    if(text[i-1] == ' ' || i == 0)
        file_input << "Yu";
    break;
case 'ю':
    if(text[i-1] == ' ' || i == 0)
        file_input << "yu";
    else
    {
        file_input << "iu";
    }
case '\ ':
    break;
case 'b':
    break;
case 'б':
    break;
default:

```

```

        file_input << text[i];
        break;
    }
}

time_t rawtime;
struct tm * timeinfo;

time (&rawtime);
timeinfo = localtime (&rawtime);

file_input << endl << "Час дозапису: " << asctime(timeinfo);

file_input.close();
}
catch(const char *error)
{
    cout << error << endl;
}
}

void task_10_3(float x, float y, float z, int b)
{
    float values[3] = {x, y, z};

    fstream file("file.txt", ios_base::out | ios_base::app);
    file.seekg(0, ios::end);
    file << "\nРезультат функції s_calculation: " << s_calculation(values) << endl <<
    "Число b: " << b;

    int bin = 0, k = 1;

    while (b)
    {
        bin += (b % 2) * k;
        k *= 10;
        b /= 2;
    }
    file << "; Число b в двійковій системі: " << bin;

    file.close();
}

```

```
}
```

Лістинг **ModulesHoi.h**:

```
#ifndef ModuleHoi
#define ModuleHoi

#include <string>

float s_calculation(int []);
std::string Beaufort_score(float);
std::string average_temperature(float [], int);
std::string amount_binary_digits(int);
void task_10_1(std::string file);
void task_10_2(std::string file);
void task_10_3(float, float, float, int);

#endif // ModulesHoi
```

Лістинг **TestDriver.cpp**:

```
#include <iostream>
#include <string>
#include "..\..\..\lab08\prj\Modules\ModulesHoi.h"

using namespace std;

int main()
{

    string f_input_10_1[2] =
    {
        "file_input_1(1).txt",
        "file_input_1(2).txt"
    };

    string f_input_10_2[2] =
    {
        "file_input_2(1).txt",
        "file_input_2(2).txt"
    };

};
```

```

int f_input_10_3[4][4] = {
    {1, 1, 1, 1},
    {2, 1, 3, 4},
    {5, 2, 11, 16},
    {3, 2, 5, 25},
};

for(int i = 0; i < size(f_input_10_3); i++)
{
    if (i < size(f_input_10_1))
        task_10_1(f_input_10_1[i]);

    if (i < size(f_input_10_2))
        task_10_2(f_input_10_2[i]);

    task_10_3(f_input_10_3[i][0],      f_input_10_3[i][1],      f_input_10_3[i][2],
f_input_10_3[i][3]);

    cout << "Натисніть Enter, щоб перейти к іншому випадку: ";
    cin.get();
}

return 0;
}

```

Назва тестового набору Test Suite Description	UnitTesting_10_1_
Назва проекту / ПЗ Name of Project / Software	ModulesHoi
Рівень тестування Level of Testing	<u>Модульний</u> / Unit Testing
Автор тест-сюита Test Suite Author	Дмитро Гой
Виконавець Implementer	Дмитро Гой

Ід-р тест- кейса / Test Case ID	Preliminary Steps	Action	Expected Result	Результат тестування (пройшов/не вдалося/ заблокований) / Test Result (passed/failed/ blocked)
1	Вміст файлу file_input_1(1).txt: Привіт! Як справи?	task_10_1(f_input_10_1[0])	Вміст файлу Ofile.txt: Розробник ПЗ: Гой Дмитро Олександрович; ЗВО: Центральноукраїнськи й національний технічний університет; Країна: Україна; Місто: Знам'янка; Рік: 2023. Кількість "?": 1 Кількість "!": 1	Passed
2	Вміст файлу file_input_1(2).txt: Првт, к спрв?	task_10_1(f_input_10_1[1])	Вміст файлу Ofile.txt: Розробник ПЗ: Гой Дмитро Олександрович; ЗВО: Центральноукраїнськи й національний технічний університет; Країна: Україна; Місто: Знам'янка; Рік: 2023. Кількість "?": 1 Кількість "!": 0 Речення в якому немає голосних: Првт, к спрв?	Passed

Назва тестового набору Test Suite Description	UnitTesting_10_2_
Назва проекту / ПЗ Name of Project / Software	ModulesHoi
Рівень тестування Level of Testing	<u>Модульний</u> / Unit Testing
Автор тест-сюїта Test Suite Author	Дмитро Гой

Ід-р тест- кейса / Test Case ID	Preliminary Steps	Action	Expected Result	Результат тестування (пройшов/не вдалося/ заблокований) / Test Result (passed/failed/ blocked)
1	Вміст файлу file_input_2(1).txt: Президентська стипендія – це стипендія, яку виплачують найкращим студентам та переможцям всеукраїнських і міжнародних олімпіад.	task_10_2(f_input_10_2[0])	Вміст файлу file_input_2(1).txt: Президентська стипендія – це стипендія, яку виплачують найкращим студентам та переможцям всеукраїнських і міжнародних олімпіад. Prezydentska stypendiia – tse stypendiia, yaku vyplachuiut naikrashchym studentam ta peremozhtsiam vseukrainskykh i mizhnarodnykh olimpiad. Час дозапису: Sun Jun 04 16:15:34 2023	Passed
2	Вміст файлу file_input_2(2).txt: Рече та стогне Дніпр широкий, сердитий вітер завива, додолу верби гне високі, горами хвилю підійма.	task_10_2(f_input_10_2[1])	Вміст файлу file_input_2(2).txt: Рече та стогне Дніпр широкий, сердитий вітер завива, додолу верби гне високі, горами хвилю підійма. Reve ta stohne Dnibr shyrokyi, serdytyi viter zavuva, dodolu verby hne vysoki, horamy khvyliu pidiima. Час дозапису: Sun Jun 04 16:19:37 2023	Passed

Назва тестового набору Test Suite Description	UnitTesting_10_3_
Назва проекту / ПЗ Name of Project / Software	ModulesHoi
Рівень тестування Level of Testing	Модульний / Unit Testing
Автор тест-сюита Test Suite Author	Дмитро Гой
Виконавець Implementer	Дмитро Гой

Ід-р тест- кейса / Test Case ID	Preliminary Steps	Action	Expected Result	Результат тестування (пройшов/не вдалося/ заблокований) / Test Result (passed/failed/ blocked)
1	Вміст файлу file.txt: (Файл порожній)	task_10_3(f_input_10_3[0][0], f_input_10_3[0][1], f_input_10_3[0][2], f_input_10_3[0][3],)	Вміст файлу file.txt: Результат функції s_calculation: nan Число b: 1; Число b в двійковій системі: 1	Passed
2	Вміст файлу file.txt: Результат функції s_calculation: nan Число b: 1; Число b в двійковій системі: 1	task_10_3(f_input_10_3[1][0], f_input_10_3[1][1], f_input_10_3[1][2], f_input_10_3[1][3],)	Вміст файлу file.txt: Результат функції s_calculation: nan Число b: 1; Число b в двійковій системі: 1 Результат функції s_calculation: 1.89483 Число b: 4; Число b в двійковій системі: 100	Passed
3	Вміст файлу file.txt: Результат функції s_calculation: nan Число b: 1; Число b в двійковій системі: 1 Результат функції	task_10_3(f_input_10_3[2][0], f_input_10_3[2][1], f_input_10_3[2][2], f_input_10_3[2][3],)	Вміст файлу file.txt: Результат функції s_calculation: nan Число b: 1; Число b в двійковій системі: 1 Результат функції s_calculation: 1.89483 Число b: 4; Число b в двійковій системі: 100 Результат функції s_calculation: 4.60859	Passed

	s_calculation: 1.89483 Число b: 4; Число b в двійковій системі: 100		Число b: 16; Число b в двійковій системі: 10000	
4	Вміст файлу file.txt: Результат функції s_calculation: nan Число b: 1; Число b в двійковій системі: 1 Результат функції s_calculation: 1.89483 Число b: 4; Число b в двійковій системі: 100 Результат функції s_calculation: 4.60859 Число b: 16; Число b в двійковій системі: 10000	task_10_3(f_input_10_3[3][0], f_input_10_3[3][1], f_input_10_3[3][2], f_input_10_3[3][3],)	Вміст файлу file.txt: Результат функції s_calculation: nan Число b: 1; Число b в двійковій системі: 1 Результат функції s_calculation: 1.89483 Число b: 4; Число b в двійковій системі: 100 Результат функції s_calculation: 4.60859 Число b: 16; Число b в двійковій системі: 10000 Результат функції s_calculation: 2.79282 Число b: 25; Число b в двійковій системі: 11001	Passed