

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

ЗВІТ
ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 9
з навчальної дисципліни
“Базові методології та технології програмування”
ПРОГРАМНА РЕАЛІЗАЦІЯ ОБРОБЛЕННЯ МАСИВІВ
ДАНИХ ТА СИМВОЛЬНОЇ ІНФОРМАЦІЇ

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

ВИКОНАВ
студент академічної групи КБ-20
Чернолес К.С.

ПЕРЕВІРИВ

ст. викладач кафедри кібербезпеки
та програмного забезпечення

Дреєва Г.М.

Кропивницький – 2024

Мета роботи : полягає у набутті ґрунтовних вмінь і практичних навичок реалізації технології модульного програмування, застосування операторів C/C++ арифметичних, логічних, побітових операцій, умови, циклів та вибору під час розроблення статичних бібліотек, заголовкових файлів та програмних засобів у кросплатформовому середовищі Code::Blocks.

1. Реалізувати функції розв’язування задач 9.1–9.3 як складових статичної бібліотеки libModulesПрізвище.a (проект ModulesПрізвище, створений під час виконання лабораторної роботи №8).

2. Реалізувати програмне забезпечення розв’язування задачі 9.4 на основі функцій статичної бібліотеки libModulesПрізвище.a.

Варіант 1

— ЗАДАЧА 9.1 —

Вхід: швидкість вітру (км/год) під час торнадо.

Вихід: категорія торнадо за шкалою Фудзіти та частота їх виникнення.



КАТЕГОРІЯ	ШВИДКІСТЬ ВІТРУ, км/год	ЧАСТОТА
F0	64 – 116	38,9 %
F1	117 – 180	35,6 %
F2	181 – 253	19,4 %
F3	254 – 332	4,9 %
F4	333 – 418	1,1 %
F5	419 – 512	менше 0,1 %

Лістинг програмного коду:

```
#include <iostream>
#include <string>
```

```
std::string getTornadoCategory(int windSpeed, double& frequency) {
```

```

    if (windSpeed < 64) {
        frequency = 0.0; // Торнадо со скоростью менее 64 км/ч не
классифицируются
        return "No tornado";
    } else if (windSpeed >= 64 && windSpeed <= 117) {
        frequency = 0.53; // Частота F0
        return "F0";
    } else if (windSpeed >= 118 && windSpeed <= 180) {
        frequency = 0.31; // Частота F1
        return "F1";
    } else if (windSpeed >= 181 && windSpeed <= 252) {
        frequency = 0.10; // Частота F2
        return "F2";
    } else if (windSpeed >= 253 && windSpeed <= 332) {
        frequency = 0.04; // Частота F3
        return "F3";
    } else if (windSpeed >= 333 && windSpeed <= 419) {
        frequency = 0.01; // Частота F4
        return "F4";
    } else if (windSpeed >= 420) {
        frequency = 0.003; // Частота F5
        return "F5";
    } else {
        frequency = 0.0; // Неопределенная частота для некорректных данных
        return "Undefined";
    }
}

int main() {
    int windSpeed;
    std::cout << "Enter wind speed during tornado (km/h): ";
    std::cin >> windSpeed;

    double frequency;
    std::string category = getTornadoCategory(windSpeed, frequency);

    if (category == "No tornado") {
        std::cout << "The given wind speed does not qualify as a tornado." <<
std::endl;
    } else {
        std::cout << "Tornado Category: " << category << std::endl;
        std::cout << "Frequency of occurrence: " << frequency << "%" <<
std::endl;
    }

    return 0;
}

```

— ЗАДАЧА 9.2 —

Вхід: температура повітря (у градусах за шкалою Цельсія), зафіксовану о 00:00, 04:00, 8:00, 12:00, 16:00, 20:00 год.

Вихід: середньодобова температура за шкалами Цельсія та Фаренгейта.



Знаючи температуру за шкалою Цельсія, температуру за шкалою Фаренгейта можна розрахувати наступним чином:

$$t_F = 32 + \frac{9}{5}t_C,$$

де t_F – температура за шкалою Фаренгейта, t_C – температура за шкалою Цельсія.

Аналіз задачі 9.2 Користувач вводить зафіксовану температуру за певний період часу, після обчислюється температура у градусах Цельсія та перетворюється на градуси Фаренгейта.

Лістинг програмного коду:

```
#include <iostream>
#include <iomanip>
#include <vector>

using namespace std;

void calculateDailyAverageTemperature(const vector<double>& temperatures) {
    double sum = 0;
    for (double temp : temperatures) {
        sum += temp;
    }
    double average = sum / temperatures.size();

    double fahrenheitAverage = average * 9 / 5 + 32;

    cout << "Daily average temperature in Celsius: " << average << " °C" <<
endl;
    cout << "Daily average temperature in Fahrenheit: " << fahrenheitAverage
<< " °F" << endl;
}

int main() {
    vector<double> temperatures;
    cout << "Enter air temperature for each hour (00:00, 04:00, 8:00, 12:00,
16:00, 20:00):" << endl;
    for (int i = 0; i < 6; ++i) {
        double temp;
        cout << "Hour " << i * 4 << ":00: ";
        cin >> temp;
        temperatures.push_back(temp);
    }
    calculateDailyAverageTemperature(temperatures);
}
```

— ЗАДАЧА 9.3 —

Вхід: натуральне число N від 0 до 65535.

Вихід: якщо біт D_0 числа N рівний 0, кількість двійкових нулів у ньому, інакше — кількість двійкових одиниць*.

*під час підрахунку кількості бінарних 0 або 1 рекомендовано використати тернарний оператор « ? : ».

Аналіз задачі 9.3 Користувач вводить число від 0 до 65535. Спочатку число потрібно перетворити в двійкову систему числення. Потім після цього порівняти 0 біт, якщо він дорівнює 0, то вивести кількість двійкових нулів, інакше – кількість двійкових одиниць.

Лістинг програмного коду:

```
#include <iostream>
#include <bitset>

int countZerosOrOnes(int N) {
    int count = 0;
    if ((N & 1) == 0) { // Check if the last bit (D0) is 0
        while (N != 0) {
            if ((N & 1) == 0) // Check if the current bit is 0
                count++;
            N >>= 1; // Right shift the number to check the next bit
        }
    } else {
        while (N != 0) {
            if ((N & 1) == 1) // Check if the current bit is 1
                count++;
            N >>= 1; // Right shift the number to check the next bit
        }
    }
    return count;
}
```

— ЗАДАЧА 9.4 —

За введенням користувачем символом “z” викликається `s_calculation()`, “r” – функція задачі 9.1, “s” – функція задачі 9.2, “t” – функція задачі 9.3; якщо користувач вводить інші символи, вони ігноруються, при чому видається звуковий сигнал про помилкове введення. Після цього, якщо користувач за запитом додатка вводить символ “q” або “Q”, відбувається вихід з програми, інакше — виконання програми повторюється.

Аналіз задачі 9.4 Користувач вводить символ с клавіатури, якщо цей символ “z” викликається `s_calculation()`, “r” – функція задачі 9.1, “s” – функція задачі 9.2, “t” – функція задачі 9.3; якщо користувач вводить інші символи, вони ігноруються, при чому видається звуковий сигнал про помилкове введення. Після цього, якщо користувач за запитом додатка вводить символ “q” або “Q”, відбувається вихід з програми, інакше виконання програми повторюється.

Лістинг програмного коду:

```
#include <iostream>
#include <vector>
#include <string>
#include <cstdlib>
#include <ctime>

using namespace std;

// Function for Problem 9.1: Tornado classification
string getTornadoCategory(int windSpeed, double& frequency) {
    if (windSpeed < 64) {
        frequency = 0.0; // Торнадо со скоростью менее 64 км/ч не
        классифицируются
        return "No tornado";
    } else if (windSpeed >= 64 && windSpeed <= 117) {
        frequency = 0.53; // Частота F0
        return "F0";
    } else if (windSpeed >= 118 && windSpeed <= 180) {
        frequency = 0.31; // Частота F1
        return "F1";
    } else if (windSpeed >= 181 && windSpeed <= 252) {
        frequency = 0.10; // Частота F2
        return "F2";
    } else if (windSpeed >= 253 && windSpeed <= 332) {
        frequency = 0.04; // Частота F3
        return "F3";
    } else if (windSpeed >= 333 && windSpeed <= 419) {
        frequency = 0.01; // Частота F4
        return "F4";
    } else if (windSpeed >= 420) {
        frequency = 0.003; // Частота F5
        return "F5";
    } else {
        frequency = 0.0; // Неопределенная частота для некорректных
        данных
        return "Undefined";
    }

    cout << "Tornado Category: " << category << std::endl;
    cout << "Frequency of occurrence: " << frequency << "%" <<
    std::endl;
}

// Function for Problem 9.2: Calculate daily average temperature
void calculateDailyAverageTemperature(const vector<double>&
temperatures) {
    double sum = 0;
    for (double temp : temperatures) {
        sum += temp;
    }
    double average = sum / temperatures.size();

    double fahrenheitAverage = average * 9 / 5 + 32;
```

```

    cout << "Daily average temperature in Celsius: " << average << "
°C" << endl;
    cout << "Daily average temperature in Fahrenheit: " <<
fahrenheitAverage << " °F" << endl;
}

```

```

// Function for Problem 9.3: Count number of binary zeros or ones
int countZerosOrOnes(int N) {
    int count = 0;
    if ((N & 1) == 0) { // Check if the last bit (D0) is 0
        while (N != 0) {
            if ((N & 1) == 0) // Check if the current bit is 0
                count++;
            N >>= 1; // Right shift the number to check the next bit
        }
    } else {
        while (N != 0) {
            if ((N & 1) == 1) // Check if the current bit is 1
                count++;
            N >>= 1; // Right shift the number to check the next bit
        }
    }
    return count;
}

```

```

// Main function where functions are called according to user input
int main() {
    char choice;
    srand(time(0)); // Initialize random number generator

    do {
        cout << "Select an option ('z' - s_calculation(), 'r' -
Problem 9.1, 's' - Problem 9.2, 't' - Problem 9.3, 'q' - quit): ";
        cin >> choice;

        switch (choice) {
            case 'z':
                // Call s_calculation() function
                cout << "Calling s_calculation() function." << endl;
                break;
            case 'r':
                int windSpeed;
                cout << "Enter wind speed (km/h): ";
                cin >> windSpeed;
                classifyTornado(windSpeed);
                break;
            case 's': {
                vector<double> temperatures;
                cout << "Enter air temperature for each hour (00:00,
04:00, 8:00, 12:00, 16:00, 20:00):" << endl;
                for (int i = 0; i < 6; ++i) {
                    double temp;
                    cout << "Hour " << i * 4 << ":00: ";
                    cin >> temp;
                    temperatures.push_back(temp);
                }
            }
        }
    } while (choice != 'q');
}

```



```

        calculateDailyAverageTemperature(temperatures);
        break;
    }
    case 't': {
        int N;
        cout << "Enter a natural number N (from 0 to 65535):
";

        cin >> N;
        int result = countZerosOrOnes(N);
        cout << "Result: " << result << endl;
        break;
    }
    case 'q':
    case 'Q':
        cout << "Exiting the program." << endl;
        break;
    default:
        cout << "\a"; // Sound alert for incorrect input
        break;
}

} while (choice != 'q' && choice != 'Q');

return 0;
}

```

Модульний тест до задачі 9.1

Назва тестового набору Test Suite Description	TS_9_1
Назва проекту / ПЗ Name of Project / Software	fudzita()
Рівень тестування Level of Testing	модульний
Автор тест-сьюта Test Suite Author	Чернолєс Кирило
Виконавець Implementer	Чернолєс Кирило

[Розрив обтікання текстом]

Ід-р тест- кейса / Test Case ID	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результат тестування (пройшов/не вдалося/ заблокований) / Test Result (passed/failed/ blocked)
TCM_1	Вхід: 100	Вихід: Tornado Category: F0 Frequency of occurrence: 0.53%	PASSED
TCM_2	Вхід: 120	Вихід: Tornado Category: F1 Frequency of occurrence: 0.31%	PASSED

TCM_3	Вхід: 435	Вихід: Tornado Category: F5 Frequency of occurrence: 0.003%	PASSED
TCM_4	Вхід: 0	Вихід: The given wind speed does not qualify as a tornado.	PASSED
TCM_5	Вхід: 200	Вихід: Tornado Category: F2 Frequency of occurrence: 0.1%	PASSED

Модульний тест до задачі 9.2

Назва тестового набору Test Suite Description	TS_9_2
Назва проекту / ПЗ Name of Project / Software	temperature_C_F()
Рівень тестування Level of Testing	модульний
Автор тест-сьюта Test Suite Author	Чернолєс Кирило
Виконавець Implementer	Чернолєс Кирило

[Розрив обтікання текстом]

Ід-р тест- кейса / Test Case ID	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результат тестування (пройшов/не вдалося/ заблокований) / Test Result (passed/failed/ blocked)
TCM_1	Вхід: Hour 0:00: 30 Hour 4:00: 10 Hour 8:00: 40 Hour 12:00: 15 Hour 16:00: 40 Hour 20:00: 20	Вихід: Daily average temperature in Celsius: 25.8333 C Daily average temperature in Fahrenheit: 78.5 F	PASSED
TCM_2	Вхід: Hour 0:00: 50 Hour 4:00: 12 Hour 8:00: 10 Hour 12:00: 23 Hour 16:00: 4 Hour 20:00: 1	Вихід: Daily average temperature in Celsius: 16.6667 C Daily average temperature in Fahrenheit: 62 F	PASSED
TCM_3	Вхід: Hour 0:00: 0 Hour 4:00: 0 Hour 8:00: 0 Hour 12:00: 0 Hour 16:00: 15 Hour 20:00: 12	Вихід: Daily average temperature in Celsius: 4.5 C Daily average temperature in Fahrenheit: 40.1 F	PASSED
TCM_4	Вхід: Hour 0:00: 0 Hour 4:00: 0 Hour 8:00: 0 Hour 12:00: 0 Hour 16:00: 0 Hour 20:00: 0	Вихід: Daily average temperature in Celsius: 0 C Daily average temperature in Fahrenheit: 32 F	PASSED

TCM_5	Вхід: Hour 0:00: 3 Hour 4:00: 4 Hour 8:00: 1 Hour 12:00: 4 Hour 16:00: 5 Hour 20:00: 6	Вихід: Daily average temperature in Celsius: 3.83333 C Daily average temperature in Fahrenheit: 38.9 F	PASSED
-------	---	--	--------

Модульний тест до задачі 9.3

Назва тестового набору Test Suite Description	TS_9_3
Назва проекту / ПЗ Name of Project / Software	binary()
Рівень тестування Level of Testing	модульний
Автор тест-сьюта Test Suite Author	Чернолєс Кирило
Виконавець Implementer	Чернолєс Кирило

[Розрив обтікання текстом]

Ід-р тест- кейса / Test Case ID	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результат тестування (пройшов/не вдалося/ заблокований) / Test Result (passed/failed/ blocked)
TCM_1	Вхід: 505	Вихід: 7	PASSED
TCM_2	Вхід: 65535	Вихід: 16	PASSED
TCM_3	Вхід: 0	Вихід: 0	PASSED
TCM_4	Вхід: 2	Вихід: Ones count: 1	PASSED
TCM_5	Вхід: 10000	Вихід: Ones count: 9	PASSED

Системний тест до задачі 9.4

Назва тестового набору Test Suite Description	TS_9_4
--	--------

Назва проекту / ПЗ Name of Project / Software	Chernoles_task
Рівень тестування Level of Testing	системний / System Testing
Автор тест-сюита Test Suite Author	Чернолес Кирило
Виконавець Implementer	Чернолес Кирило

[Розрив обтікання текстом]

Ід-р тест- кейса / Test Case ID	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результат тестування (пройшов/не вдалося/ заблокований) / Test Result (passed/failed/ blocked)
TS-1	<ol style="list-style-type: none"> Ввести "r" Ввести 200 Ввести "q" 	Enter action ==>r Enter wind speed (km\h) ==>200 Out ==> Tornado Category: F2 Frequency of occurrence: 0.1% ==>q	passed
TS-2	<ol style="list-style-type: none"> Ввести "s" Ввести "10" Ввести -8 9 7 -6 0 -4 -7 -1 -2 -10 Ввести "q" 	Enter action ==>s Enter air temperature for each hour (00:00, 04:00, 8:00, 12:00, 16:00, 20:00) ==>30, 10, 40, 15, 40, 20 Out: ==> Daily average temperature in Celsius: 25.8333 C Daily average temperature in Fahrenheit: 78.5 F Enter action ==>q	passed
TS-3	<ol style="list-style-type: none"> Ввести "t" Ввести 1000 Ввести "q" 	Enter action ==>t Enter num ==>1000 Out ==> Result: 9 Enter action ==>q	passed
TS-4	<ol style="list-style-type: none"> Ввести "f" Ввести "j" Ввести "q" 	Enter action ==>f *Звуковий сигнал* Enter action ==>j *Звуковий сигнал* Enter action ==>q	passed