

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
МЕХАНІКО-ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ
КАФЕДРА КІБЕРБЕЗПЕКИ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ЗВІТ
з лабораторної роботи № 9
з дисципліни
«Базові методології та технології програмування»
на тему:
РЕАЛІЗАЦІЯ ПРОГРАМНИХ МОДУЛІВ РОЗГАЛУЖЕНИХ ТА ІТЕРАЦІЙНИХ
ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

Виконав:
студент академічної групи КН-22
ПІВНЕНКО О.М.

Перевірів:
викладач
СОБІНОВ О.Г.

Лабораторна робота № 9

Варіант № 1

ТЕМА: Реалізація програмних модулів розгалужених та ітераційних обчислювальних процесів.

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

ЗАВДАННЯ:

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 %

9.2 Вхід температура повітря (у градусах за шкалою Цельсія), зафіксовану о 00:00, 04:00, 08:00, 12:00, 16:00, 20:00 год. Вихід: середньодобова температура за шкалою Цельсія та Фаренгейта.

9.3 Вхід: натуральне число N від 0 до 65535. Вихід: якщо біт D₀ числа N рівний 0, кількість двійкових нулів у ньому, інакше – кількість двійкових одиниць.

9.4. За введеними 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.

ХІД ЛАБОРАТОРНОЇ РОБОТИ

Слід зауважити, що вихідний код проектів TestDriver та ModulesPivnenko наведений у додатках 1, 2 відповідно. Тест-сюти до завдань 9.1, 9.2, 9.3, 9.4 наведені у додатках 3, 4, 5, 6.

Завдання 9.1

Проалгоритмізуємо задачу й візьмемо допущення. Швидкість вітру (speedWind) – ціле число. На виході ми повинні отримати частоту й певну категорію торнадо, тому доцільним буде створити для вирішення задачі структуру, яка буде містити два поля: тип торнадо, частота виникнення (перше ми опишемо за допомогою «enum», а друге – це число із плаваючою комою, тому float). Вихідний код отриманої функції зображений у лістингу 9.1 (а також є частиною додатку 2, що представляє собою увесь код ModulesPivnenko). Вихідний код структури, який використовується, а також перелічення (enum) наведені у лістингу 9.2.

Лістинг 9.1.

```
tornado_result task_9_1(int speedWind){
    tornado_result desc; desc.freq = 0; desc.tornType = ERR_F;

    if(speedWind >= 64 && speedWind <= 116){
        desc.freq = 38.9;
        desc.tornType = category::F0;
    }
    else if(speedWind >= 117 && speedWind <= 180){
        desc.freq = 35.6;
        desc.tornType = category::F1;
    }
    else if(speedWind >= 181 && speedWind <= 253){
        desc.freq = 19.4;
        desc.tornType = category::F2;
    }
    else if(speedWind >= 254 && speedWind <= 332){
        desc.freq = 4.9;
        desc.tornType = category::F3;
    }
    else if(speedWind >= 333 && speedWind <= 418){
        desc.freq = 1.1;
        desc.tornType = category::F4;
    }
    else if(speedWind >= 419 && speedWind <= 512){
```

```

        desc.freq = 0.1;
        desc.tornType = category::F5;
    }
    return desc;
}

```

Лістинг 9.2.

```

enum category{
    F0,
    F1,
    F2,
    F3,
    F4,
    F5,
    ERR_F
};

struct tornado_result{
    category tornType;
    float freq;
};

```

Завдання 9.2

Алгоритмізуємо задачу: на вході ми отримуємо 6 чисел із плаваючою комою (в градусах Цельсія), опісля ми повинні на виході отримати середньодобову температуру (це, з великою ймовірністю, теж число з плаваючою комою) у градусах Цельсія й за шкалою Фаренгейта. Тут також було б доречно створити структуру, що містила б обидві змінні. Вихідний код синтезованої, алгоритмізованої задачі зображено у лістингу 9.3. Вихідний код структури, що використовується, зображений у лістингу 9.4.

Лістинг 9.3.

```

temperatureDay task_9_2(float temp1, float temp2, float temp3, float temp4,
float temp5, float temp6){
    float averageVal = (temp1 + temp2 + temp3 + temp4 + temp5 + temp6)/6;

    temperatureDay outTemperatures;

    outTemperatures.byCelsius = averageVal;
    outTemperatures.byFahrenheit = (32.0 + (9.0/5.0)*averageVal);

    return outTemperatures;
}

```

Лістинг 9.4.

```
struct temperatureDay{
    float byCelsius;
    float byFahrenheit;
};
```

Завдання 9.3

Проведемо алгоритмізацію задачі. На вході ми отримуємо натуральне число від 0 до 65536. В такому разі доречним буде використовувати тип `unsigned short` (який приймає значення у цьому діапазоні). Якщо перший біт числа `N` дорівнює 0, то ми повинні кількість двійкових нулів, якщо дорівнює 1, то повинні вивести кількість одиниць. Доречним буде на виході отримати певне число типу `int` (хоча, треба зауважити, що можна було б взяти `char` задля економії пам'яті, але декілька байтів – це не суттєва навантаження на сучасні ПК). Вихідний код функції наведений у лістингу 9.5. Варто також зауважити, що у цій функції використовуються допоміжні функції. Вони зображені теж у лістингу 9.5.

Лістинг 9.5.

```
int task_9_3(unsigned short N){
    // Нехай біт D0 - це перший біт. Наприклад нехай маємо ^0010010010010011
    // Тоді в такому разі біт після символу ^ буде вважатись бітом D0.

    char counter = 0;
    unsigned short mask = 0b1000000000000000;

    int result = N & mask;

    if(result == 0){
        counter = calc_bits(N, 0);
    }
    else if(result > 0){
        counter = calc_bits(N, 1);
    }
    else {
        cout << "Something is bad...";
        return -1;
    }

    return counter;
}

int calc_bits(unsigned short N, char bit){
```

```

char cnt_bits = 0;
char cnt = sizeof(N) * 8; // bits
char i = 0;
unsigned short mask = 0b0100000000000000;

if(bit == 0){
    while(i < cnt){
        if((N & mask) == bit){
            cnt_bits += 1;
        }
        mask = mask >> 1;
        i += 1;
    }
}
else if(bit == 1){
    while(i < cnt){
        if((N & mask) >= bit){
            cnt_bits += 1;
        }
        mask = mask >> 1;
        i += 1;
    }
    cnt_bits += 1;
}

return cnt_bits;
}

```

Завдання 9.4.

Алгоритмізуємо задачу: на вході ми отримуємо певний символ, а після перевіряємо, чи він є одним із тих, після котрого ми повинні викликати функцію. Якщо так, то викликаємо потрібну функцію (це залежить від символу), якщо ні, то подаємо звуковий сигнал й чекаємо на повторне введення символу. Вихідний код додатку зображений у лістингу 9.6.

Лістинг 9.6.

```

#include <iostream>
#include "ModulesPivnenko.hpp"

using namespace std;

int main()
{

```

```

char symbol = '\0';

while(symbol != 'Q' && symbol != 'q'){
    cin >> symbol;

    if(symbol == 'z'){
        float result = s_calculation(1, 1, 1);
        cout << "s_calculation result is " << result << "." << endl;
    }
    else if(symbol == 'r'){
        tornado_result obj = task_9_1(100);
        cout << "Frequency and category: " << obj.freq << ", " << obj.tornType
<< endl;
    }
    else if(symbol == 's'){
        temperatureDay obj = task_9_2(10, 15, 20, 25, 30, 35);
        cout << "Average temperature by Celsius and Fahrenheit: " <<
obj.byCelsius << ", " << obj.byFahrenheit << "." << endl;
    }
    else if(symbol == 't'){
        cout << "Count on nulls ( or ones ) in number, " << task_9_3(100) <<
"." << endl;
    }
    else{
        cout << "\a";
    }

}

return 0;
}

```

ВИСНОВОК

У ході виконання лабораторної роботи я закріпив свої знання (а також здобув нові) з мови програмування C++. Попрацював, а також вдосконалив свої навички роботи із текстовими редакторами Sublime Text, Notepad++, IDE Code::Blocks, IDE Visual Studio. Закріпив свої знання із тестування програмного забезпечення, циклів, арифметичних, логічних, порозрядних операцій у C++. Також створив більш функціональну бібліотеку на вищеназваній мові. Вирішив

задачі на визначення категорії торнадо, обчислення середньодобової температури, переведення значення із градусів Цельсія в градуси по шкалі Фаренгейта. Попрацював із двійковою системою представлення чисел. Варто також зазначити особисті враження від виконання лабораторної роботи: я отримав лише позитивні емоції: створення більш серйозних, функціональних додатків наближує мене до того, щоб стати кваліфікованим фахівцем у сфері комп'ютерних наук.

Вихідний код TestDriver:

```

#include <iostream>
#include <locale.h>
#include <windows.h>
#include <cmath>
#include "ModulesPivnenko.hpp"

using namespace std;

float shortVal(float in_val){
    in_val = round(in_val*100)/100;

    return in_val;
}

bool testCase_9_1(int speedWind, category etalon){
    tornado_result result = task_9_1(speedWind);

    if(result.tornType == etalon)
        return true;
    return false;
}

bool testCase_9_2(float temp1, float temp2, float temp3, float temp4, float temp5, float
temp6, float etalonCelsie, float etalonFahr){
    temperatureDay result = task_9_2(temp1, temp2, temp3, temp4, temp5, temp6);

    result.byCelsius = shortVal(result.byCelsius);
    result.byFahrenheit = shortVal(result.byFahrenheit);
    etalonCelsie = shortVal(etalonCelsie);
    etalonFahr = shortVal(etalonFahr);

    cout << "Celsius val: " << result.byCelsius << endl;
    cout << "Fahrenheit val: " << result.byFahrenheit << endl;

    if(result.byCelsius == etalonCelsie && result.byFahrenheit == etalonFahr)
        return true;
    return false;
}

bool testCase_9_3(unsigned short N, int etalon){
    int cnt_bits = task_9_3(N);

    if(cnt_bits == etalon){
        return true;
    }
    return false;
}

bool TestSuite_9_1(void){
    int speedVals[8] = { 70, 125, 200, 300, 367, 421, -5, 600};
    category catVal[8] = { F0, F1, F2, F3, F4, F5, ERR_F, ERR_F};

    bool findFail = false;

    for(int i = 0; i < 8; i += 1){
        bool test_res = testCase_9_1(speedVals[i], catVal[i]);
        cout << "TC number " << i << ": " << speedVals[i] << ". Category: " << catVal[i] <<
endl;
        cout << "Result is ";
        if(test_res == true)

```

```

        cout << "PASSED." << endl;
    if(test_res == false){
        findFail = true;
        cout << "FAILED" << endl;
    }
}

if(findFail == true)
    return false;

return true;
}

bool TestSuite_9_2(void){
    bool resultSuite = true;

    float temp_vals[7][6] =
    {
        { 20, 21, 23, 24, 25, 26 },
        { 10, 15, 20, 25, 30, 40},
        { -10, -20, -30, -40, -50, -60},
        { -4, -7, -11, -1, -8, -20 },
        { 5.5, 10.1, 20.1, 9.9, 30.5, 31 },
        { -7.1, -3.5, -9.253, -1, -11.2, -10 },
        { 0, 0, 0, 0, 1, 1 }
    };

    float etalon_vals[7][2] =
    {
        { 23.166666, 73.7 },
        { 23.33333, 73.9999 },
        { -35, -31 },
        { -8.5, 16.7},
        { 17.8501, 64.13 },
        { -7.01000023, 19.3799992 },
        { 0.33, 32.6 }
    };

    for(int i = 0; i < 7; i += 1){
        bool case_res = testCase_9_2(temp_vals[i][0], temp_vals[i][1], temp_vals[i][2],
temp_vals[i][3], temp_vals[i][4], temp_vals[i][5], etalon_vals[i][0], etalon_vals[i][1]);

        cout << "Input values: " << temp_vals[i][0] << ", " << temp_vals[i][1] << ", ";
        cout << temp_vals[i][2] << ", " << temp_vals[i][3] << ", " << temp_vals[i][4] << ",
" << temp_vals[i][5] << endl;
        cout << "Etalon values:" << etalon_vals[i][0] << ", " << etalon_vals[i][1] << endl;

        if(case_res == true)
            cout << "Result is PASSED." << endl;
        else{
            cout << "Result is FAILED." << endl;
            resultSuite = false;
        }
    }
    return resultSuite;
}

bool TestSuite_9_3(void){
    unsigned short in_vals[10] = {0, 100, 200, 2340, 65535, 21000, 7801, 42050, 1, 50000};
    int benchmark[10] =
        {16, 13, 13, 12, 16, 12, 7, 5, 15, 6 };
    // int benchark_ones[10] =
        { 0, 3, 3, 4, 16, 6, 9, 6, 1, 5
    };

    for(int i = 0; i < 10; i += 1){
        if(testCase_9_3(in_vals[i], benchmark[i]) == true)

```

```
        cout << in_vals[i] << ", " << benchmark[i] << ". " << "PASSED." << endl;
    else
        cout << in_vals[i] << ", " << benchmark[i] << ". " << "FAILED." << endl;
}

return false;
}

int main()
{
    cout << "TEST DRIVER." << endl << "DEVELOPER: OLEKSANDR PIVNENKO, KN-22, MTF" << endl
    << "LABORATORY WORK 9.\n\n\n";

    TestSuite_9_1();
    cout << "\n\n";

    TestSuite_9_2();
    cout << "\n\n";

    TestSuite_9_3();

    return 0;
}
```

Вихідний код ModulesPivnenko.

Файл main.cpp:

```
#include <math.h>
#include "ModulesPivnenko.hpp"
#define PI 3.1415
#include <iostream>

using namespace std;

int calc_bits(unsigned short N, char bit){
    char cnt_bits = 0;
    char cnt = sizeof(N) * 8; // bits
    char i = 0;
    unsigned short mask = 0b0100000000000000;

    if(bit == 0){
        while(i < cnt){
            if((N & mask) == bit){
                cnt_bits += 1;
            }
            mask = mask >> 1;
            i += 1;
        }
    }
    else if(bit == 1){
        while(i < cnt){
            if((N & mask) >= bit){
                cnt_bits += 1;
            }
            mask = mask >> 1;
            i += 1;
        }
        cnt_bits += 1;
    }

    return cnt_bits;
}

float s_calculation(int x, int y, int z){
    float pow_res = pow(2*z + 1, x);

    float sqrt_res = sqrt(abs(y - ((float)1/(float)2) * z));

    float S = pow_res - sqrt_res + z + PI;

    return S;
}

tornado_result task_9_1(int speedWind){
    tornado_result desc; desc.freq = 0; desc.tornType = ERR_F;

    if(speedWind >= 64 && speedWind <= 116){
        desc.freq = 38.9;
        desc.tornType = category::F0;
    }
    else if(speedWind >= 117 && speedWind <= 180){
        desc.freq = 35.6;
        desc.tornType = category::F1;
    }
    else if(speedWind >= 181 && speedWind <= 253){
        desc.freq = 19.4;
    }
}
```

```

        desc.tornType = category::F2;
    }
    else if(speedWind >= 254 && speedWind <= 332){
        desc.freq = 4.9;
        desc.tornType = category::F3;
    }
    else if(speedWind >= 333 && speedWind <= 418){
        desc.freq = 1.1;
        desc.tornType = category::F4;
    }
    else if(speedWind >= 419 && speedWind <= 512){
        desc.freq = 0.1;
        desc.tornType = category::F5;
    }
    }

    return desc;
}

temperatureDay task_9_2(float temp1, float temp2, float temp3, float temp4, float temp5,
float temp6){
    float averageVal = (temp1 + temp2 + temp3 + temp4 + temp5 + temp6)/6;

    temperatureDay outTemperatures;

    outTemperatures.byCelsius = averageVal;
    outTemperatures.byFahrenheit = (32.0 + (9.0/5.0)*averageVal);

    return outTemperatures;
}

int task_9_3(unsigned short N){
    // Нехай біт D0 - це перший біт. Наприклад нехай маємо ^0010010010010011
    // Тоді в такому разі біт після символу ^ буде вважатись бітом D0.

    char counter = 0;
    unsigned short mask = 0b1000000000000000;

    int result = N & mask;

    if(result == 0){
        counter = calc_bits(N, 0);
    }
    else if(result > 0){
        counter = calc_bits(N, 1);
    }
    else {
        cout << "Something is bad...";
        return -1;
    }

    return counter;
}

```

Заголовковий файл:

```

#ifndef MODULESPIVNENKO_HPP_INCLUDED
#define MODULESPIVNENKO_HPP_INCLUDED

enum category{
    F0,
    F1,
    F2,
    F3,

```

```
    F4,  
    F5,  
    ERR_F  
};  
  
struct tornado_result{  
    category tornType;  
    float freq;  
};  
  
struct temperatureDay{  
    float byCelsius;  
    float byFahrenheit;  
};  
  
float s_calculation(int x, int y, int z);  
  
tornado_result task_9_1(int speedWind);  
  
temperatureDay task_9_2(float, float, float, float, float, float);  
  
int task_9_3(unsigned short N);  
  
#endif // MODULESPIVNENKO_HPP_INCLUDED
```

Artifact: Test Suite

Date: 4/13/2023

Назва тестового набору Test Suite Description	TS_lab9_1
Назва проекту / ПЗ Name of Project / Software	ModulesPivnenko
Рівень тестування Level of Testing	системний / System Testing
Автор тест-сьюта Test Suite Author	Олександр Півненко
Виконавець Implementer	Олександр Півненко

Ід-р тест-кейса / Test Case ID	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результат тестування / Test Result
TC_01	Вхідні дані: число з діапазону [64, 116]	Вихідні дані: категорія торнадо F0	PASSED
TC_02	Вхідні дані: число з діапазону [117, 180]	Вихідні дані: категорія торнадо F1	PASSED
TC_03	Вхідні дані: число з діапазону [181, 253]	Вихідні дані: категорія торнадо F2	PASSED
TC_04	Вхідні дані: число з діапазону [254, 332]	Вихідні дані: категорія торнадо F3	PASSED
TC_05	Вхідні дані: число з діапазону [333, 418]	Вихідні дані: категорія торнадо F4	PASSED
TC_06	Вхідні дані: число з діапазону [419, 512]	Вихідні дані: категорія торнадо F5	PASSED
TC_07	Вхідні дані: число з діапазону < 0	Вихідні дані: категорія торнадо ERR_F	PASSED

Додаток 4.

Artifact: Test Suite

Date: 4/13/2023

Назва тестового набору Test Suite Description	TS_lab9_2
Назва проекту / ПЗ Name of Project / Software	ModulesPivnenko
Рівень тестування Level of Testing	системний / System Testing
Автор тест-сюита Test Suite Author	Олександр Півненко
Виконавець Implementer	Олександр Півненко

Ід-р тест- кейса / Test Case ID	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результат тестування / Test Result
TC_01	Вхідні дані: 20, 21, 23, 24, 25, 26	Вихідні дані: 23,1(6) за Цельсієм, 73,6(9) за Фаренгейтом	PASSED
TC_02	Вхідні дані: 10, 15, 20, 25, 30, 40	Вихідні дані: 23,3(3) за Цельсієм, 73,9(9) за Фаренгейтом	PASSED
TC_03	Вхідні дані: -10, -20, -30, -40, -50, -60	Вихідні дані: -35 за Цельсієм, -31 за Фаренгейтом	PASSED
TC_04	Вхідні дані: -4, -7, -11, -1, -8, -20	Вихідні дані: -8,5 за Цельсієм, 16,7 за Фаренгейтом	PASSED
TC_05	Вхідні дані: 5,5; 10,1; 20,1; 9,9, 30,5; 31	Вихідні дані: 17,85 за Цельсієм, 64,13 за Фаренгейтом.	PASSED
TC_06	Вхідні дані: -7,1; -3,5; -9,253; -1; -11,2; -10	Вихідні дані: -7,0088(3) за Цельсієм, 19,38 за Фаренгейтом.	PASSED
TC_07	Вхідні дані: 0, 0, 0, 0, 1, 1	Вихідні дані: 0,3(3) за Цельсієм, 32,6 за Фаренгейтом.	PASSED

Додаток 5.

Artifact: Test Suite

Date: 4/13/2023

Назва тестового набору Test Suite Description	TS_lab9_3
Назва проекту / ПЗ Name of Project / Software	ModulesPivnenko
Рівень тестування Level of Testing	системний / System Testing
Автор тест-сьюта Test Suite Author	Олександр Півненко
Виконавець Implementer	Олександр Півненко

Ід-р тест- кейса / Test Case ID	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результат тестування / Test Result
TC_01	Вхідні дані: 0	Вихідні дані: 16	PASSED
TC_02	Вхідні дані: 100	Вихідні дані: 13	PASSED
TC_03	Вхідні дані: 200	Вихідні дані: 13	PASSED
TC_04	Вхідні дані: 2340	Вихідні дані: 12	PASSED
TC_05	Вхідні дані: 65535	Вихідні дані: 16	PASSED
TC_06	Вхідні дані: 21000	Вихідні дані: 12	PASSED
TC_07	Вхідні дані: 7801	Вихідні дані: 7	PASSED
TC_08	Вхідні дані: 42050	Вихідні дані: 5	PASSED
TC_09	Вхідні дані: 1	Вихідні дані: 15	PASSED
TC_10	Вхідні дані: 50000	Вихідні дані: 6	PASSED

Додаток 6.

Artifact: Test Suite

Date: 4/13/2023

Назва тестового набору Test Suite Description	TS_lab9_4
Назва проекту / ПЗ Name of Project / Software	ModulesPivnenko
Рівень тестування Level of Testing	системний / System Testing
Автор тест-сьюта Test Suite Author	Олександр Півненко
Виконавець Implementer	Олександр Півненко

Ід-р тест- кейса / Test Case ID	Дії (кроки) / Action (Test Steps)	Очікуваний результат / Expected Result	Результат тестування / Test Result
TC_01	1. Відкрити Pivnenko_task.exe 2. Увести q	Вікно програми закрите.	PASSED
TC_02	1. Відкрити Pivnenko_task.exe 2. Увести r 3. Увести q	Після введення r ми побачимо: Frequency and category: 38.9, 0 Опісля q вікно застосунку закрите.	PASSED
TC_03	1. Відкрити Pivnenko_task.exe 2. Увести rt 3. Увести q	Після введення rq ми побачимо: Frequency and category: 38.9, 0 Count on nulls (or ones) in number, 13. Опісля q вікно застосунку закрите.	PASSED
TC_04	1. Відкрити Pivnenko_task.exe 2. Увести rtq	Вікно застосунку закрите	PASSED
TC_05	1. Відкрити Pivnenko_task.exe 2. Увести s	Вікно застосунку відкрите. На екрані надпис: Average temperature by Celsius and Fahrenheit: 22.5, 72.5.	PASSED
TC_06	1. Відкрити Pivnenko_task.exe 2. Увести k	Вікно застосунку відкрите. Чекає на введення нового символу. Після введення k відтворено звуковий сигнал.	PASSED
TC_07	1. Відкрити Pivnenko_task.exe 2. Увести k 3. Натиснути ENTER 4. Увести st	Вікно застосунку відкрите. Надпис, який міститься у ньому: Average temperature by Celsius and Fahrenheit: 22.5, 72.5. Count on nulls (or ones) in number, 13.	PASSED
TC_08	1. Відкрити Pivnenko_task.exe 2. Натиснути ENTER.	Вікно застосунку відкрите. Звуковий сигнал відсутній. Програма очікує введення символу.	PASSED
TC_09	1. Відкрити Pivnenko_task.exe 2. Увести p 3. Увести k 4. Увести q	Під час введення p та k були відтворені два звукових сигнали. Опісля введення q вікно застосунку закрите.	PASSED
TC_10	1. Відкрити Pivnenko_task.exe 2. Увести Qsr	Вікно застосунку закрите.	PASSED