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”

РЕАЛІЗАЦІЯ ПРОГРАМНИХ МОДУЛІВ РОЗГАЛУЖЕНИХ ТА ІТЕРАЦІЙНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

ЗАВДАННЯ ВИДАВ

доцент кафедри кібербезпеки та програмного забезпечення

Доренський О. П.

[https://github.com/odorenskyi/](https://github.com/odorenskyi/Dmytro-Parkhomenko-KB18)

ВИКОНАВ

студент академічної групи КБ-24 Олейніков О. С

ПЕРЕВІРИВ викладач кафедри кібербезпеки та програмного забезпечення

Коваленко А. С.

Кропивницький – 2025

# ЛАБОРАТОРНА РОБОТА №9

**Тема:** Реалізація програмних модулів розгалужених та ітераційних обчислювальних процесів.

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

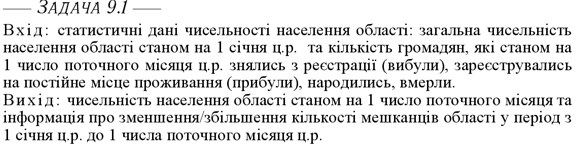
**Варіант №21**

**Завдання:**

1. Реалізувати функції розв’язування задач 9.1–9.3 як складових статичної бібліотеки libModulesПрізвище.а (проект ModulesПрізвище, створений під час виконання лабораторної роботи №8).
2. Реалізувати програмне забезпечення розв’язування задачі 9.4 на основі функцій статичної бібліотеки libModulesПрізвище.а.

**Хід Роботи**

**Завдання 9.1**



**Строга постанова задачі**

Вхідні данні: чисельність населення області, кількість смертельних випадків та кількість новонароджених, кількість імігрантів та емігрантів (ціле число більше за нуль), станом на 1 число поточного місяця.

Вихідні данні: Чисельність населення з урахуванням всіх змінних та число зміни населення після обрахування

**Лістинг модуля задачі 9.1**

population\_flow changes(int population,int death,int birth,int emigraation,int imigration){

struct population\_flow changes;

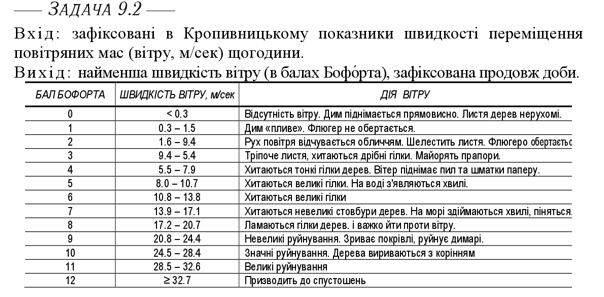
changes.flow = birth - death + imigration - emigraation;

changes.population = population + changes.flow;

return changes;

}

**Завдання 9.2**



**Строга постанова задачі**

Вхідні данні: швидкість вітру що година на протязі останніх 24 годин (десяткове число більше нуля)

Вихідні данні: Сила вітру у Балах Бофорта найменшой швидкості вітру на протязі останніх 24 годин

**Лістинг Модуля задачі 9.2**

short int bal\_boforta(float wind[24]){

float Min = wind[0];

for(int i = 1;i<24;i++){

if (wind[i] < Min){

Min = wind[i];

}

}

if (Min < 0.3){

return 0;

}

else if (Min < 1.5){

return 1;

}

else if (Min < 3.4){

return 2;

}

else if (Min < 5.4){

return 3;

}

else if (Min < 7.9){

return 4;

}

else if (Min < 10.7){

return 5;

}

else if (Min < 13.8){

return 6;

}

else if (Min < 17.1){

return 7;

}

else if (Min < 20.7){

return 8;

}

else if (Min < 24.4){

return 9;

}

else if (Min < 28.4){

return 10;

}

else if (Min < 32.6){

return 11;

}

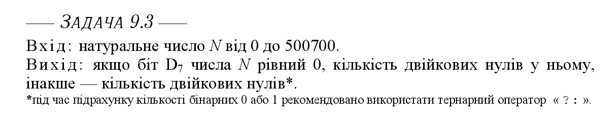
else {

return 12;

}

}

**Завдання 9.3**



**Строга постанова задачі**

Вхідні данні:Натуральне число в проміжку від 0 до 500700

Вихідні данні: якщо число по 7 індексу = 0 то кількість двійкових одиниць, Якщо число = 1 то кількість двійкових одиниць

**Лістинг модуля задачі 9.3**

short int binary\_func(unsigned int num){

int zeros = 0;

int ones = 0;

bool return\_zeros = false;

for (int i = 0; num > 0;i++){

if (i == 7){

if (num%2 == 0){

return\_zeros = true;

}

}

if (num%2 == 1){

ones++;

}

else {

zeros++;

}

num >>= 1;

}

if (return\_zeros){

return zeros;

}

else{

return ones;

}

}

**TESTDRIVER**

1) Пдіключаємо статичну бібліотеку «Modules\_Oleinikov.a» та заголовковий файл Modules\_Oleinikov.h.

2) Тестуємо очікуваний результат кожної функції (задач 9.1 – 9.3).

**Лістинг тестового драйвера задач 9.1 – 9.3**

#include <iostream>

#include "Modules-oleinikov.h"

using namespace std;

void test\_binary\_func()

{

unsigned int possibleresult[5] = {13,11,10,5,4};

unsigned int numm[5] = {655304,500403,40378,1688,3264};

cout << "binary function test" << endl;

for (short int i = 0; i < 5; i++)

{

if (binary\_func(numm[i]) == possibleresult[i]){

cout << "Test #[" << i + 1 << "]: PASSED\n";

}

else{

cout << "Test #[" << i + 1 << "]: FAILED\n";

}

}

}

void test\_bofort\_ball()

{

int expected\_res[5] = {4,0,5,2,1};

float wind\_power0[24] = {14.1,8.3,6.2,7.2,32.1,8.1,7.5,8.9,12.7,16.4,29.2, 15.7, 26.6, 8.5, 36.1, 18.8, 20.7, 31.7, 17.9, 34.3, 27.9, 26.5, 15.8, 16.2};

float wind\_power1[24] = {0.6, 6.1, 7.4, 9.1, 11.4, 11.6, 11.8, 13.8, 15.0, 0.2, 20.9, 24.7, 25.0, 25.4, 26.3, 27.3, 28.7, 29.6, 30.2, 31.7, 32.5, 33.4, 36.8, 38.1};

float wind\_power2[24] = {25.1, 33.7, 23.4, 27.2, 29.5, 36.9, 39.1, 16.2, 32.8, 33.6, 31.7, 37.6, 36.3, 36.6, 40.0, 19.9, 21.9, 25.8, 13.2, 11.8, 21.3, 9.5, 33.1, 18.9};

float wind\_power3[24] = {12.6, 34.4, 10.7, 9.0, 13.0, 7.2, 7.9, 18.0, 10.0, 33.0, 35.2, 26.8, 10.4, 22.2, 28.4, 20.0, 24.3, 18.8, 22.5, 21.1, 1.5, 9.3, 27.7, 33.5};

float wind\_power4[24] = {32.2, 5.0, 1.2, 3.0, 9.8, 3.4, 23.2, 24.0, 32.3, 22.3, 27.7, 27.4, 7.5, 37.7, 0.4, 30.2, 15.1, 25.8, 14.4, 27.3, 37.2, 23.0, 0.3, 13.3};

cout << "Bafort ball test" << endl;

if (bal\_boforta(wind\_power0) == expected\_res[0]){

cout << "Test #[1]: PASSED\n";

}

else{

cout << "Test #[1]: FAILED\n";

}

if (bal\_boforta(wind\_power1) == expected\_res[1]){

cout << "Test #[2]: PASSED\n";

}

else{

cout << "Test #[2]: FAILED\n";

}if (bal\_boforta(wind\_power2) == expected\_res[2]){

cout << "Test #[3]: PASSED\n";

}

else{

cout << "Test #[3]: FAILED\n";

}if (bal\_boforta(wind\_power3) == expected\_res[3]){

cout << "Test #[4]: PASSED\n";

}

else{

cout << "Test #[4]: FAILED\n";

}if (bal\_boforta(wind\_power4) == expected\_res[4]){

cout << "Test #[5]: PASSED\n";

}

else{

cout << "Test #[5]: FAILED\n";

}

}

void test\_population\_flow()

{

int population[5] = {1000,3500,8000,16000,8432};

int death[5] = {121,567,221,168,632};

int birth[5] = {100,344,567,872,888};

int imigration[5] = {500,400,342,343,200};

int emigration[5] = {577,880,901,457,789};

population\_flow possibleres[5] = {{1056,56},{3757,257},{8905,905},{16818,818},{9277,845}};

cout << "Population flow function test" << endl;

for (short int i = 0; i < 5; i++)

{

if (changes(population[i],death[i],birth[i],imigration[i],emigration[i]).population == possibleres[i].population &&

changes(population[i],death[i],birth[i],imigration[i],emigration[i]).flow == possibleres[i].flow){

cout << "Test #[" << i + 1 << "]: PASSED\n";

}

else{

cout << "Test #[" << i + 1 << "]: FAILED\n";

}

}

}

int main()

{

test\_population\_flow();

test\_bofort\_ball();

test\_binary\_func();

}

**Результат роботи TestDriver.exe**

Population flow function test

Test #[1]: PASSED

Test #[2]: PASSED

Test #[3]: PASSED

Test #[4]: PASSED

Test #[5]: PASSED

Bafort ball test

Test #[1]: PASSED

Test #[2]: PASSED

Test #[3]: PASSED

Test #[4]: PASSED

Test #[5]: PASSED

binary function test

Test #[1]: PASSED

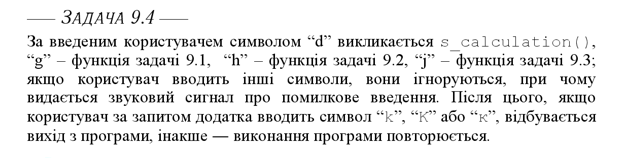
Test #[2]: PASSED

Test #[3]: PASSED

Test #[4]: PASSED

Test #[5]: PASSED

**Завдання 9.4**



**Алгоритмізація задачі 9.4**

1)Підключаємо статичну бібліотеку «Modules-oleinikov.a» та заголовковий файл «Modules-oleinikov.h»

2)Вводимо символ

3)Якщо символ не співпадає з перелічиними символами повертаємося на крок 2

4)Якщо введений символ дорівнює «к», «K» або «k» - завершення програми

5)Якщо символ = d то

5.1)Об`являємо 3 цілочисельні змінні

5.2)Вводимо 3 числа

5.3)Ввиводимо результат функції s\_calculation

6)Якщо символ = g

6.1)Об’являємо 5 цілочисельні змінні

6.2)Вводим 5 чисел

6.3)Виводим результат функції population\_flow\_changes

7)Якщо символ = h

7.1)Оголошуємо масив на 24 елементи

7.2) Вводимо 24 елементи

7.3)Виводимо результат функції bal\_boforta

8)Якщо символ = j

8.1)Оголошуємо цілочисельну зміну

8.2)Водимо число в діапазоні від 0 до 500700

8.3)Виводимо результат функції binary\_func

9)Користувач може вводити всі перечислені символи до поки не введе «k», «K» або «к» для завершення програми

**Лістинг програми 9.4**

#include <iostream>

#include <conio.h>

#include "Modules-oleinikov.h"

using namespace std;

int main()

{

system("chcp 1251 & cls");

int x,y,z;

float wind\_power[24];

int population,death,birth,imigration,emigration;

int numm;

cout << "d - calls a function s\_calculation" << endl

<< "j - calls a binary function" << endl

<< "h - calls a bofort\_bal function" << endl

<< "g - calls a population\_flow function" << endl

<< "'k','K' and 'к' terminate the program" << endl;

getchar();

while(true) {

cout << "Enter symbol (d/j/h/g/k/K/к): ";

switch (getch()) {

case 'd': cout << endl;

cout << "Enter x value: ";

cin >> x;

cout << "Enter y value: ";

cin >> y;

cout << "Enter z value: ";

cin >> z;

cout << "the result of expression S = " << s\_calculation(x,y,z) << endl;

break;

case 'j': cout << endl;

cout << "Enter number in range 0 - 500700: ";

cin >> numm;

cout << "Tne amount of binary zeros or onces is " << binary\_func(numm) << endl;

cout << endl;

break;

case 'h': cout << endl;

for(int i = 0;i<24;i++){

cout << "Enter wind power per hour of 24 hours: ";

cin >> wind\_power[i];

}

cout << "Beafort bal of the weakest wind power in 24 hours is " << bal\_boforta(wind\_power);

cout << endl;

break;

case 'g': cout << endl;

cout << "Enter population of the region: ";

cin >> population;

cout << "Enter number of birth on the region: ";

cin >> birth;

cout << "Enter number of death cases of the region: ";

cin >> death;

cout << "Enter number of emigration cases in the region: ";

cin >> emigration;

cout << "Enter number of imigration cases in the region: ";

cin >> imigration;

cout << "Curent number of population in the region is " << changes(population,birth,death,emigration,imigration).population << endl;

cout << "Curent number of changes in the region is " << changes(population,birth,death,emigration,imigration).flow << endl;

cout << endl;

break;

case 'k': return 0; break;

case 'K': return 0; break;

case 'к': return 0; break;

default: cout << "\a" << endl << endl; continue;

}

}

}

**Висновок:**

Під час виконання **Лабораторної роботи №9** я зіткнувся з низкою труднощів. Наприклад, під час тестування функції **bal\_boforta** спочатку планував використати **двовимірний масив**, але через проблеми з реалізацією цієї ідеї замінив його на **5 одновимірних масивів**. В результаті код вийшов об’ємнішим, ніж очікувалося.

Також виникли **складності з підключенням бібліотек та файлів** до основної частини програми. Щоб виправити це, довелося змінити розташування файлів і налаштування компілятора.

У **завданні 9.4** я згадав, як працює конструкція **switch**, і навчився використовувати її разом із функцією **getch**. Хоча спочатку мені було незрозуміло, як правильно її застосовувати, згодом я освоїв цей підхід.

Найбільш **складним** для мене виявилося **написання модуля для завдання 9.2**, оскільки я не відразу зрозумів, як правильно задати умови для всіх швидкостей вітру. Через це довелося кілька разів переписувати цей фрагмент коду.

У **тест-драйвері** я вирішив **не виводити порівняння результатів** з очікуваними значеннями, оскільки це зробило б код громіздким. Також мені було **незручно вводити 24 елементи масиву вручну** через консоль — цей процес займав багато часу і вимагав постійного придумування чисел.

**Завдання 9.1–9.3** були відносно простими, але й вони мали свої нюанси: **умови циклів, різні типи змінних, перетворення даних**. Наприклад, я стикався з помилками компілятора через дублювання змінних типу **float**, і лише після кількох спроб код запрацював.

Ще одна проблема виникла з **локалізацією** в **завданні 9.4**: програма мала завершуватися при натисканні символів **«K», «k» або «к»**, але компілятор замінив **«к»** на невідомий символ. Після зміни типу локалізації проблема зникла.

Найбільше часу зайняло **написання тест-сьютів** для кожної частини програми та **exe-файлу**. Хоча саме завдання було нескладним, процес виявився досить тривалим.

**Реалізація модулів** для кожного завдання теж зайняла багато часу, але ці зусилля виправдалися: після написання **тест-драйверів** я знаходив менше помилок, особливо в **завданні 9.4**. Цього разу я вирішив спочатку створювати **модулі та тести**, а потім переходити до основного коду — це допомогло уникнути зайвих витрат часу на виправлення помилок.

Найскладнішим для мене стало **завдання 9.3** — попри просту умову, я не відразу зрозумів, як перевіряти **7-й індекс числа**. Однак після кількох спроб я знайшов правильний підхід. Також багато часу пішло на функцію **population\_flow\_changes**: я пробував різні методи, але працездатним виявився лише один — той, що зараз є в бібліотеці.

Загалом, робота була цікавою, але вимагала багато уваги до деталей.

**Додаток А**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_9\_1 |
| Назва проекта / ПЗ  Name of Project / Software | population\_flow\_changes() |
| Рівень тестування  Level of Testing | модульний |
| Автор тест-сьюта  Test Suite Author | Олейніков Олександр |
| Виконавець  Implementer | Олейніков Олександр |

|  |  |  |  |
| --- | --- | --- | --- |
| 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:1000,121,100,500,577 | Вихід: 1056,56 | PASSED |
| TCM\_2 | Вхід:3500,567,344,400,880 | Вихід: 3757,257 | PASSED |
| TCM\_3 | Вхід: 8000,221,567,342,901 | Вихід: 8905,905 | PASSED |
| TCM\_4 | Вхід: 16000,168,872,343,457 | Вихід: 16818,818 | PASSED |
| TCM\_5 | Вхід: 8432,632,888,200,789 | Вихід: 9277,845 | PASSED |

**Додаток Б**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_9\_2 |
| Назва проекта / ПЗ  Name of Project / Software | bofort\_ball() |
| Рівень тестування  Level of Testing | модульний |
| Автор тест-сьюта  Test Suite Author | Олейніков Олександр |
| Виконавець  Implementer | Олейніков Олександр |

|  |  |  |  |
| --- | --- | --- | --- |
| 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: 14.1,8.3,6.2,7.2,32.1,8.1,7.5,8.9,12.7,16.4,29.2, 15.7, 26.6, 8.5, 36.1, 18.8, 20.7, 31.7, 17.9, 34.3, 27.9, 26.5, 15.8, 16. | Вихід: 4 | PASSED |
| TCM\_2 | Вхід: 0.6, 6.1, 7.4, 9.1, 11.4, 11.6, 11.8, 13.8, 15.0, 0.2, 20.9, 24.7, 25.0, 25.4, 26.3, 27.3, 28.7, 29.6, 30.2, 31.7, 32.5, 33.4, 36.8, 38.1 | Вихід: 0 | PASSED |
| TCM\_3 | Вхід: 25.1, 33.7, 23.4, 27.2, 29.5, 36.9, 39.1, 16.2, 32.8, 33.6, 31.7, 37.6, 36.3, 36.6, 40.0, 19.9, 21.9, 25.8, 13.2, 11.8, 21.3, 9.5, 33.1, 18.9 | Вихід: 5 | PASSED |
| TCM\_4 | Вхід: 12.6, 34.4, 10.7, 9.0, 13.0, 7.2, 7.9, 18.0, 10.0, 33.0, 35.2, 26.8, 10.4, 22.2, 28.4, 20.0, 24.3, 18.8, 22.5, 21.1, 1.5, 9.3, 27.7, 33.5 | Вихід: 2 | PASSED |
| TCM\_5 | Вхід: 32.2, 5.0, 1.2, 3.0, 9.8, 3.4, 23.2, 24.0, 32.3, 22.3, 27.7, 27.4, 7.5, 37.7, 0.4, 30.2, 15.1, 25.8, 14.4, 27.3, 37.2, 23.0, 0.3, 13.3 | Вихід: 1 | PASSED |

**Додаток В**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_9\_3 |
| Назва проекта / ПЗ  Name of Project / Software | binary\_func() |
| Рівень тестування  Level of Testing | модульний |
| Автор тест-сьюта  Test Suite Author | Олейніков Олександр |
| Виконавець  Implementer | Олейніков Олександр |

|  |  |  |  |
| --- | --- | --- | --- |
| 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: 655304 | Вихід: 13 | PASSED |
| TCM\_2 | Вхід:500403 | Вихід: 11 | PASSED |
| TCM\_3 | Вхід: 40378 | Вихід: 10 | PASSED |
| TCM\_4 | Вхід: 1688 | Вихід: 5 | PASSED |
| TCM\_5 | Вхід: 3264 | Вихід: 4 | PASSED |

**Додаток Г**

|  |  |
| --- | --- |
| Назва тестового набору  Test Suite Description | TS\_9\_4 |
| Назва проекта / ПЗ  Name of Project / Software | Oleinikov\_task |
| Рівень тестування  Level of Testing | системний / System Testing |
| Автор тест-сьюта  Test Suite Author | Олейніков Олександр |
| Виконавець  Implementer | Олейніков Олександр |

|  |  |  |  |
| --- | --- | --- | --- |
| 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 | 1. Запустити програму | d - calls a function s\_calculation  j - calls a binary function  h - calls a bofort\_bal function  g - calls a population\_flow function  'k','K' and 'к' terminate the program | PASSED |
| TS-2 | 1. Ввести “t” 2. Ввести d 3. Ввести 3 4. Ввести 1 5. Ввести 2 | d - calls a function s\_calculation  j - calls a binary function  h - calls a bofort\_bal function  g - calls a population\_flow function  'k','K' and 'к' terminate the program  Enter symbol (d/j/h/g/k/K/к):  \*Звуковий сигнал\*  Enter symbol (d/j/h/g/k/K/к):  Enter x value: 3  Enter y value: 1  Enter z value: 2  the result of expression S = -4.10265  Enter symbol (d/j/h/g/k/K/к): | PASSED |
| TS-3 | 1. Ввести “j” 2. Ввести 64320 3. Ввести “k” | d - calls a function s\_calculation  j - calls a binary function  h - calls a bofort\_bal function  g - calls a population\_flow function  'k','K' and 'к' terminate the program  Enter symbol (d/j/h/g/k/K/к):  Enter number in range 0 - 500700: 64320  Tne amount of binary zeros or onces is 8  \*Закриття додатку\* | PASSED |
| TS-4 | 1. Ввести “d” 2. Ввести 16000 3. Ввести 873 4. Ввести 432 5. Ввести 250 6. Ввести 320 7. Ввести “j” 8. Ввести 432081 | d - calls a function s\_calculation  j - calls a binary function  h - calls a bofort\_bal function  g - calls a population\_flow function  'k','K' and 'к' terminate the program  Enter symbol (d/j/h/g/k/K/к):  Enter symbol (d/j/h/g/k/K/к):  Enter population of the region: 16000  Enter number of birth on the region: 873  Enter number of death cases of the region: 432  Enter number of emigration cases in the region: 250  Enter number of imigration cases in the region: 320  Curent number of population in the region is 15629  Curent number of changes in the region is -371  Enter symbol (d/j/h/g/k/K/к):  Enter number in range 0 - 500700: 432081  Tne amount of binary zeros or onces is 11 | PASSED |
| TS-5 | 1. Ввести “h” 2. Ввести 1.2, 0.1, 5.9, 31.0, 16.4, 11.1, 10.2, 0.8, 22.6, 21.1, 11.9, 20.1, 13.1, 18.4, 19.5, 27.8, 16.9, 23.4, 21.9, 30.1, 11.3, 12.1, 21.0, 20.2 3. Ввести “к” | d - calls a function s\_calculation  j - calls a binary function  h - calls a bofort\_bal function  g - calls a population\_flow function  'k','K' and 'к' terminate the program  Enter symbol (d/j/h/g/k/K/к):  Enter wind power per hour of 24 hours: 1.2  Enter wind power per hour of 24 hours: 0.1  Enter wind power per hour of 24 hours: 5.9  Enter wind power per hour of 24 hours: 31.0  Enter wind power per hour of 24 hours: 16.4  Enter wind power per hour of 24 hours: 11.1  Enter wind power per hour of 24 hours: 10.2  Enter wind power per hour of 24 hours: 0.8  Enter wind power per hour of 24 hours: 22.6  Enter wind power per hour of 24 hours: 21.1  Enter wind power per hour of 24 hours: 11.9  Enter wind power per hour of 24 hours: 20.1  Enter wind power per hour of 24 hours: 13.1  Enter wind power per hour of 24 hours: 18.4  Enter wind power per hour of 24 hours: 19.5  Enter wind power per hour of 24 hours: 27.8  Enter wind power per hour of 24 hours: 16.9  Enter wind power per hour of 24 hours: 23.4  Enter wind power per hour of 24 hours: 21.9  Enter wind power per hour of 24 hours: 30.1  Enter wind power per hour of 24 hours: 11.3  Enter wind power per hour of 24 hours: 12.1  Enter wind power per hour of 24 hours: 21.0  Enter wind power per hour of 24 hours: 20.2  Beafort bal of the weakest wind power in 24 hours is 0  \*Закриття програми\* | PASSED |