МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

“ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ УПРАВЛІННЯ

ЗВІТ

З ЛАБОРАТОРНОЇ РОБОТИ №5

ЗА ТЕМОЮ “КОМБІНАТОРНІ АЛГОРИТМИ”

Виконав студент

групи КН-221д

Кукуєв Руслан Олександрович

Перевірив

Солонська С.В.

Харків 2022

ЛАБОРАТОРНА РОБОТА №5

Тема: КОМБІНАТОРНІ АЛГОРИТМИ.

Мета роботи: познайомитися з комбінаторними алгоритмами.

**Завдання:**

Розробити програму, яка читає з клавіатури число N (1 < N < 256) та параметри генератору випадкових чисел та виводить на екран послідовність з N згенерованих чисел. Програма зберігає до файлу графічну характеристику послідовності згідно завдання та виводить на екран результат одного з тестів NIST (згідно варіанту завдання). Варіанти генераторів випадкових чисел. 1 Лінійний конгруентний метод. 2 Метод Фібоначчі із затримуванням. Варіанти графічних характеристик. 1 Гістограма розподілу елементів послідовності. 2 Розподіл на площині (елементи попарно обробляються як координати точок (x, y)). 3 Автокореляція (користувач задає зсув для копії послідовності). Варіанти тестів NIST. 1 Частотний побітовий тест. 2 Тест на послідовність однакових бітів. 3 Тест на найдовшу послідовність одиниць.

**Варіант №12**

#include <iostream>

#include <fstream>

#include <time.h>

using namespace std;

unsigned int lagA = 55, lagB = 24;

void GenerateRandomValues(int N, int \*Nums);

int MaxConsecutiveOnes(int N, int \*Nums);

void WritingToFile(int N, int \*Nums);

int main()

{

int N = 0;

int \*Nums = new int[N];

cout << "Enter the number of generated values: ";

cin >> N;

if (1 >= N || N > 256)

{

std::cout << "Incorrect value: " << N << "\n";

system("pause");

return 0;

}

GenerateRandomValues(N, Nums);

WritingToFile(N, Nums);

cout << "Test max consecutive ones: " << MaxConsecutiveOnes(N, Nums) << "\n";

system("pause");

}

void GenerateRandomValues(int N, int \*Nums)

{

srand(time(NULL));

int \*Arr = new int[max(lagA, lagB)];

for (int i = 0; i < max(lagA, lagB); i++)

{

Arr[i] = rand();

}

for (int i = N; i > 0; i--)

{

int Res;

if (Arr[max(lagA, lagB) - lagA] >= Arr[max(lagA, lagB) - lagB])

{

Res = Arr[max(lagA, lagB) - lagA] - Arr[max(lagA, lagB) - lagB];

}

else

{

Res = Arr[max(lagA, lagB) - lagB] - Arr[max(lagA, lagB) - lagA];

}

for (int i = 0; i < max(lagA, lagB); i++)

{

Arr[i] = rand();

}

Nums[i] = Res;

}

}

int MaxConsecutiveOnes(int N, int \*Nums)

{

int maxValue = 0;

for (int i = 0; i < N; i++)

{

int count = 0;

while (Nums[i] != 0)

{

Nums[i] = (Nums[i] & (Nums[i] << 1));

count++;

}

if (count > maxValue)

{

maxValue = count;

}

}

return maxValue;

}

void WritingToFile(int N, int \*Nums)

{

std::ofstream out;

out.open("E:\\Projects\\C++\\OTA Labs\\CharacteristicSequence.txt");

if (out.is\_open())

{

out << "Characteristic sequence:\n";

for (int i = 1; i < N; i++)

{

out << i << " : Value: " << Nums[i] << "\n";

}

}

else

{

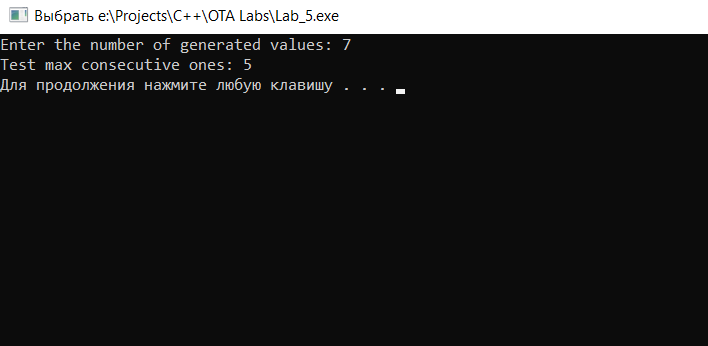
std::cout << "Open (out)file error\n";

}

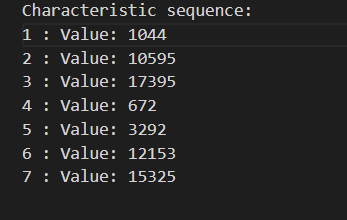
out.close();

}

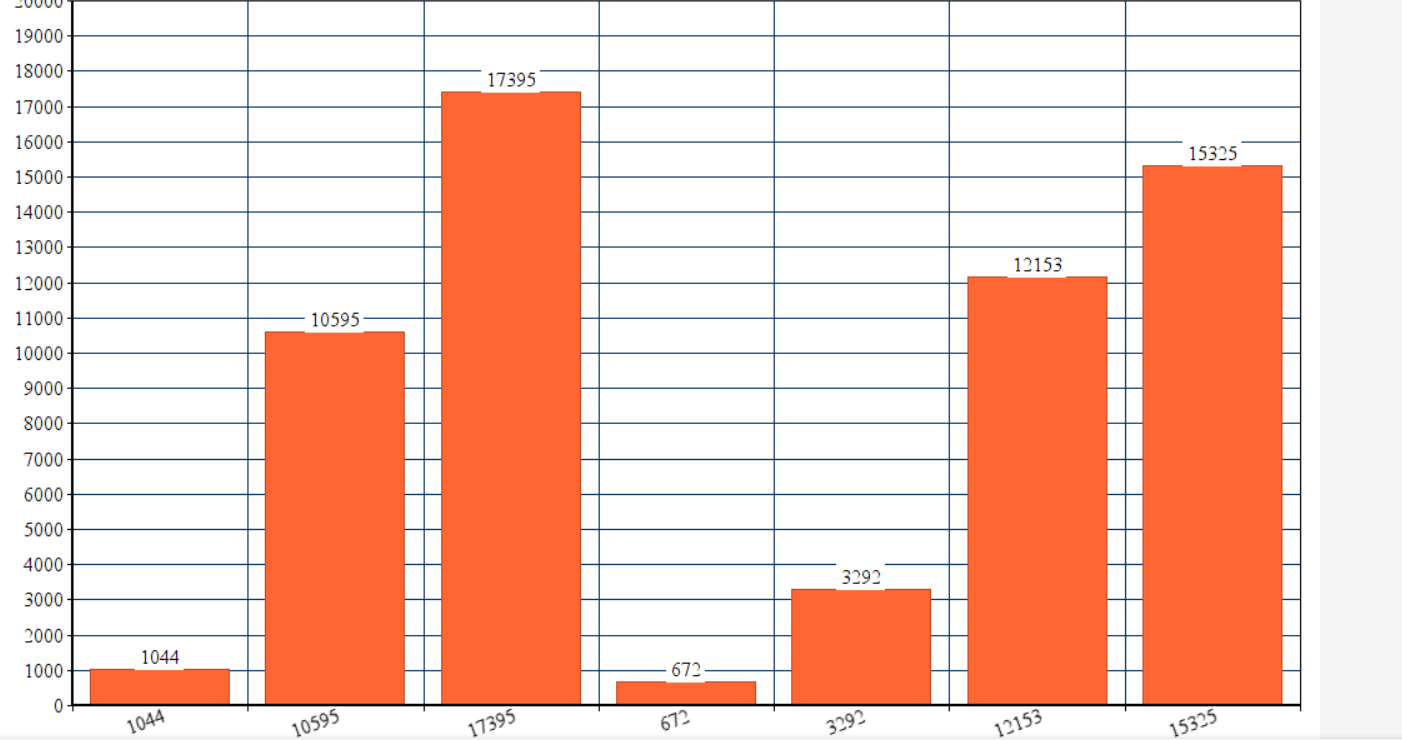
**Демонстрація роботи:**



**Вихідні дані:**



**Гістограма:**



**Висновок:** виконавши лабораторну роботу №5, я познайомився з комбінаторними алгоритмами.