МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра програмної інженерії та інформаційних технологій управління

Звіт з лабораторної роботи № 5

з дисципліни «Основи теорії алгоритмів»

Виконав: студент гр. КН-221Д

Габорець

Перевірила:

Солонська С.В.

Харків

2022

ЛАБОРАТОРНА РОБОТА №5

***Тема:*** *КОМБІНАТОРНІ АЛГОРИТМИ*

***Мета роботи:*** *познайомитися з генераторами випадкових чисел та методами перевірки випадковості.*

***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:***

*Лінійний конгруентний метод.*

***Графічна характеристика:***

Автокореляція (користувач задає зсув для копії послідовності).

**Тест NIST:**

Частотний побітовий тест.



Значення(графік) автокореляції у текстовому файлі, файл :



Код:

#include <iostream>

#include <fstream>

#include <math.h>

using namespace std;

void Autocore(int\* arr, int n)//характеристика последовательности//Автокореляция

{

double sum = 0, sum1 = 0;

double result[300];

int offcet;

cout << "Enter an offset value for the autocorrelation function:" << std::endl;

cin >> offcet;

if (offcet > 256)

{

cout << "Too much offset!\n";

exit(0);

}

ofstream File("res.txt");

for (int i = 1; i <= n + offcet; i++)

{

for (int j = 1; j <= n - 1; j++)

{

if (offcet + j + i <= n)

sum += arr[j] \* arr[abs(offcet + i + j)];//abs-модуль

sum1 += arr[j] \* arr[j];

}

result[i] = sum / sum1;

sum = 0;

sum1 = 0;

}

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

{

File << i << " Iteration.....: " << result[i] << " " << "\n";

}

}

void Nist\_Test(int\* arr, int n)//Частотний побітовий тест.

{

double h = 0, res = 0;

int g = 0;

int secArr[8192]; // 256\*32 = 8192 bit

int result[8192];

for (int i = 1; i <= n \* 32; ++i)//32 bit

{

secArr[i] = (arr[i] >> i) & 1;

}

for (int i = 1; i <= n \* 32; i++)

{

result[i] = (2 \* secArr[i] - 1);

g += result[i];

}

h = g / sqrt(2 \* n);

res = erfc(h);//erfc( x ) возвращает Дополнительную Функцию ошибок, оцененную для каждого элемента x

cout << "NIST\_bit test = " << res << "\n";

}

int main()

{

cout << "Enter the count of nums of array:" << "\n";

int n;

cin >> n;

if (n > 256 || n < 1)

{

cout << "Error, 256< N <1\n";

return -1;

}

int m = 245, c = 17, a = 3; //Линейн.конгр.метод

int array[256];

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

{

array[i] = ((array[i - 1] \* a) + c) % m;//x[n+1]=(a\*X[n]+c)% m

}

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

{

cout << "array[" << i << "] = " << array[i] << "\n";

}

cout << "\n";

Autocore(array, n);

cout << "\n";

Nist\_Test(array, n);

return 0;

}

**Висновок**: виконавши лабораторну роботу №5, я отримав навички роботи з генераторами випадкових чисел та методами перевірки випадковості та отримав навички програмування алгоритмів, що їх обробляють.