`

Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

Лабораторная работа №3

По дисциплине «Криптографические методы защиты информации»

Тема: «Генерирование равномерно распределенных псевдослучайных последовательностей»

Выполнил:

Студент 2 курса

Группы ИИ-21

Литвинюк Т. В.

Проверил:

Хацкевич М. В.

Брест 2023

`

Цель: освоить основные алгоритмы программного генерировния псевдослучайных последовательностей.

Ход работы:

Вариант 6

5	Конгруэнтный генератор со случайными параметрами	N=15	*Рассмотреть следующие случаи:* 1. $B = \left\{\begin{pmatrix}3\\0\end{pmatrix}, \begin{pmatrix}3\\1\end{pmatrix}, \begin{pmatrix}3\\2\end{pmatrix}\right\}$ 2. B выбрать произвольно, $\lvert B\rvert = 5$

CongruentGet.h:

```
#include <vector>
using namespace std;

class CongruentGen {
public:
    CongruentGen(vector<vector<unsigned long long>> &args, unsigned long long n){
        this->args = args;
        this->n = n;
    }

    unsigned long long gen() {
        if (t == args.size())
            t = 0;
        a = args[t][0];
        b = args[t++][1];
        x = (a * x + b) % n;
        return x;
    }

private:
    unsigned long long a;
    unsigned long long b;
    unsigned long long n;
    unsigned long long x = 1;
    unsigned t = 0;
    vector<vector<unsigned long long>> args;
};
```

task.cpp:

```
#include <iostream>
#include "./CongruentGen.h"
using namespace std;

int main() {
    cout << "B = {{3, 0}, {3, 1}, {3, 2}}, n = 15:\n";
    vector<vector<unsigned long long>> args1 = {{3, 0}, {3, 1}, {3, 2}};
    CongruentGen generator1(args1, 15);
    for (int i = 0; i < 20; i++) {
        cout << generator1.gen() << " ";
    }
    cout << "\n============================================================\n";

    cout << "a = 3, c = 1, n = 15:\n";
    vector<vector<unsigned long long>> args2 = {{25, 17}, {13, 11}, {17, 19}, {23, 7}, {29, 13}};
    CongruentGen generator2(args2, 15);
    for (int i = 0; i < 30; i++) {
        cout << generator2.gen() << " ";
    }
}
```

```
B = {{3, 0}, {3, 1}, {3, 2}}, n = 15:
3 10 2 6 4 14 12 7 8 9 13 11 3 10 2 6 4 14 12 7
============================================================
{{25, 17}, {13, 11}, {17, 19}, {23, 7}, {29, 13}}, n = 15:
12 2 8 11 2 7 12 13 6 7 12 2 8 11 2 7 12 13 6 7 12 2 8 11 2 7 12 13 6 7
```

Вывод: в ходе лабораторной работы я научился создавать алгоритм генерации псевдо случайной последовательности.