

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АДЫГЕЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Инженерно-физический факультет
Кафедра автоматизированных систем обработки информации и
управления

ОТЧЕТ ПО ПРАКТИКЕ

Вариант 2

2 курс, группа 2УТС-30

Выполнил:

_____ М. В. Климов
«___» _____ 2022 г.

Руководитель:

_____ С. В. Теплоухов
«___» _____ 2022 г.

Майкоп, 2022 г.

1. Введение

- 1) Текстовая формулировка задачи
- 2) код данной задачи
- 3) скриншот программы

2. Вариант 2

задание

Генератор случайных чисел Парка-Миллера с перетасовкой и без перетасовки.

Теория

Самая простая последовательность, которая можно предложить для реализации генератора равномерного распределения:

$$I(j+1) = a * I(j) \pmod{m}$$

при соответствующей выборке констант. Константы были предложены Park и Miller:

$$a = 7^5 = 16807, \quad m = 2^{31} - 1 = 2147483647.$$

Модуль разлагается в выражение:

$$m = a * q + r$$

Если $r < q$ и $0 < z < m-1$, то при этом величины $a * (z \bmod q)$ и $r * [z/q]$ всегда лежат в интервале $0, \dots, m-1$. Для умножения $(a * z) \pmod{m}$ при этом используется алгоритм:

- $t = a(z \bmod q) - r[z/q]$

- если $t < 0$, то $t += m$.

- $(a * z) \pmod{m} = t$

В случае констант Парка-Миллера можно использовать $q = 12773$ и $r = 2836$.

3. Ход работы

3.1. Код приложения

```
#include <stdio.h>
#define IA 16807
#define IM 2147483647
#define AM (1./IM)
#define IQ 12773
#define IR 2836
#define NTAB 32
#define NWUP 8
```

```

#define NDIV (1+(IM-1)/NTAB)
#define EPS 1.2e-7
#define RNMX (1.0-EPS)
#define MASK 123456789

static long dummy;
void Seed(long dum) {
dummy = dum;
}

float unirand0(void) {
long k;
float ans;

dummy ^= MASK;
k = dummy / IQ;

if ((dummy = IA * (dummy - k * IQ) - IR * k) < 0)
dummy += IM;

ans = AM * dummy;

dummy ^= MASK;

return(ans);
}

float unirand1(void) {
int j;
long k;
static long iy = 0, iv[NTAB];
float temp;

if (dummy <= 0 || !iy) {
if (dummy < 0) dummy = -dummy; else
if (dummy == 0) dummy = 1;
for (j = NTAB + NWUP - 1; j >= 0; j--) {
k = dummy / IQ;

if ((dummy = IA * (dummy - k * IQ) - IR * k) < 0)
dummy += IM;

if (j < NTAB)
iv[j] = dummy;

```

```

}

iy = iv[0];
}

k = dummy / IQ;
if ((dummy = IA * (dummy - k * IQ) - IR * k) < 0)
dummy += IM;

iy = iv[j = iy / NDIV];
iv[j] = dummy;

if ((temp = AM * iy) > RNMX)
return(RNMX);
else
return(temp);
}

int main() {
int i;
Seed(6723);
for (i = 0; i < 100; i++)
printf("%f\n", unirand1());
}

//Генератор случайных чисел Парка-Миллера без перетасовки

/*#include <stdio.h>
#define IA 16807
#define IM 2147483647
#define AM (1./IM)
#define IQ 12773
#define IR 2836
#define MASK 123456789

static long dummy;
void Seed(long dum) {
dummy = dum;
}

float unirand0(void) {
long k;
float ans;

```

```

dummy ^= MASK;
k = dummy / IQ;

if ((dummy = IA * (dummy - k * IQ) - IR * k) < 0)
dummy += IM;

ans = AM * dummy;

dummy ^= MASK;

return(ans);
}

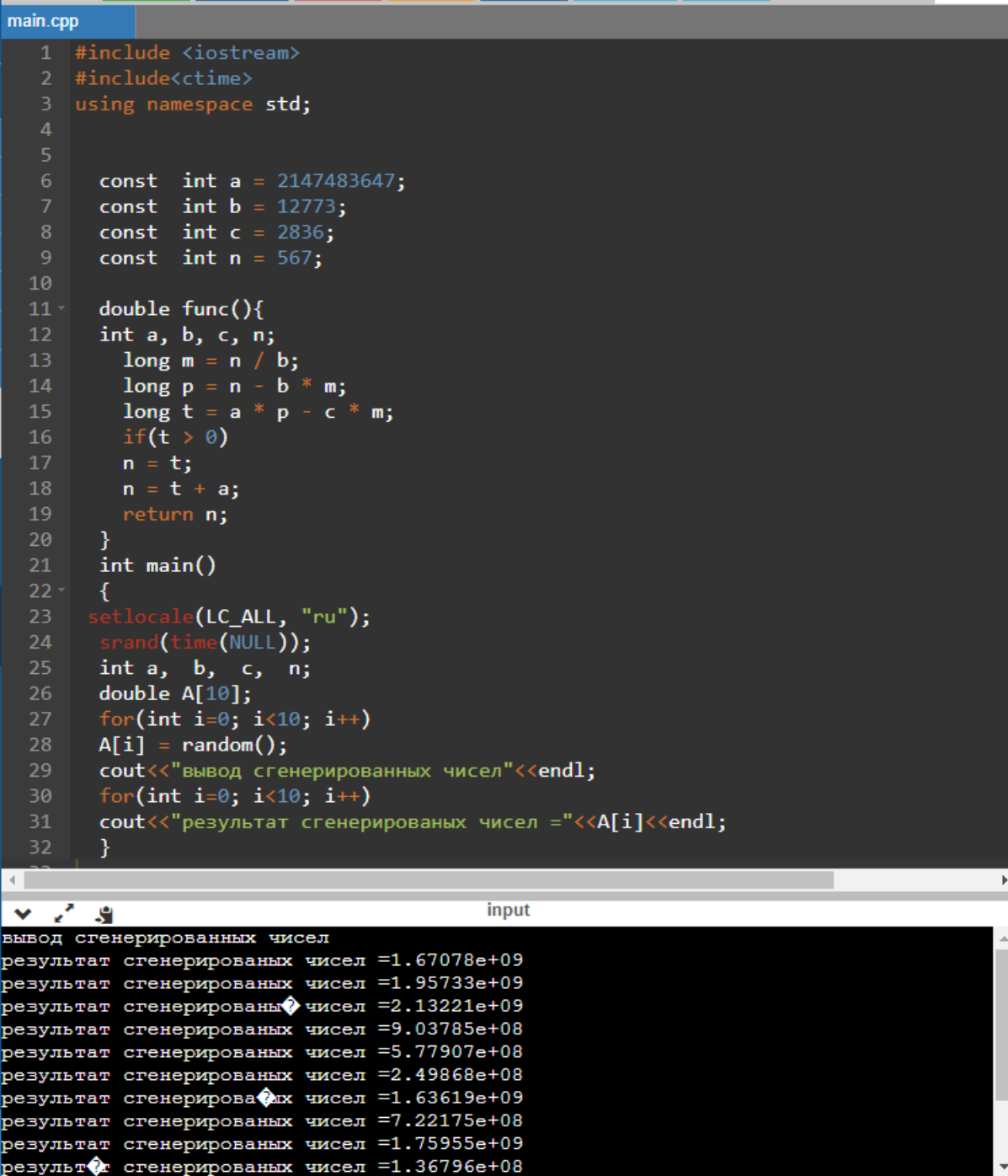
int main() {
int i;
Seed(6723);
for (i = 0; i < 100; i++)
printf("%f\n", unirand0());
}*/

```

3.2. формулы

Общая формула генератора случайных чисел $X_{k+1} = a * X_k \bmod m$

4. Скриншот программы



The screenshot shows a C++ program in a code editor and its execution output in a terminal window. The code defines constants a, b, and c, and a function func() that calculates a value based on these constants and a parameter n. The main function generates 10 random numbers and prints them along with the result of the func() function for each number.

```
main.cpp
1  #include <iostream>
2  #include<ctime>
3  using namespace std;
4
5
6  const int a = 2147483647;
7  const int b = 12773;
8  const int c = 2836;
9  const int n = 567;
10
11 double func(){
12     int a, b, c, n;
13     long m = n / b;
14     long p = n - b * m;
15     long t = a * p - c * m;
16     if(t > 0)
17         n = t;
18     n = t + a;
19     return n;
20 }
21 int main()
22 {
23     setlocale(LC_ALL, "ru");
24     srand(time(NULL));
25     int a, b, c, n;
26     double A[10];
27     for(int i=0; i<10; i++)
28         A[i] = random();
29     cout<<"вывод сгенерированных чисел"<<endl;
30     for(int i=0; i<10; i++)
31         cout<<"результат сгенерированных чисел ="<<A[i]<<endl;
32 }
```

input

```
вывод сгенерированных чисел
результат сгенерированных чисел =1.67078e+09
результат сгенерированных чисел =1.95733e+09
результат сгенерированных чисел =2.13221e+09
результат сгенерированных чисел =9.03785e+08
результат сгенерированных чисел =5.77907e+08
результат сгенерированных чисел =2.49868e+08
результат сгенерированных чисел =1.63619e+09
результат сгенерированных чисел =7.22175e+08
результат сгенерированных чисел =1.75955e+09
результат сгенерированных чисел =1.36796e+08
```

1. Скриншот программы.

5. библиографические ссылки

Для изучения «внутренностей» $\text{T}_\text{E}\text{X}$ необходимо изучить [1], а для использования $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ лучше почитать [2, 3].

Список литературы

- [1] Кнут Д.Э. Всё про $\text{T}_\text{E}\text{X}$. — Москва: Изд. Вильямс, 2003 г. 550 с.
- [2] Львовский С.М. Набор и верстка в системе $\text{L}_\text{A}\text{T}_\text{E}\text{X}$. — 3-е издание, исправленное и дополненное, 2003 г.
- [3] Воронцов К.В. $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ в примерах. 2005 г.