# 算法设计与分析实验报告

# 实验四 随机数产生器

|  |  |
| --- | --- |
| **院系：** | **信息工程学院** |
| **班级：** | **15级网络工程2班** |
| **学号：** | **2015551621** |
| **姓名：** | **王 康** |
| **任课教师：** | **王 婷** |
| **成绩：** |  |

**湘 潭 大 学**

**2018年6月实验四 随机数产生器**

1. **实验内容**

设计一个随机数发生器，可以产生分布在任意整数区间[a,b]的随机数序列。

**二．实验目的**

1、掌握线性同余法产生随机数的方法；

2、了解计算机中的随机数是如何产生的，以及为什么将随机数称为伪随机数。

**三. 算法描述**

1、采用递推式（线性同余）： Tn = (A\*Tn-1+B) mod C ，通过设置A、B、C

三个参数，易知可以产生0~C之间的序列。

2、采用设置种子的方法，用全局变量保存种子，然后代入到递推式中，产生一个伪随机数，用全局变量保存，作为产生下一个伪随机数的种子。

**四. 算法实现**

1. **数据结构及函数说明**

**Ⅰ、数据结构**

全局变量randNum保存种子和上一个产生的伪随机数。

**Ⅱ、函数说明**

/\*\*线性同余： Tn = (A\*Tn-1+B) mod C 易知Tn 为 0~C 之间的数

\* @author Swking

\* @method pseudoRand

\* @parame

\* @return randNum 伪随机数

\*/

double pseudoRand(){

randNum = ((randNum\*9301)%49853 + 49297)%49999 ;

return randNum;

}

1. **源程序代码**

#include<iostream>

#include<algorithm>

#include<cstdio>

#include<ctime>

#include<bits/stdc++.h>

using namespace std;

int randNum;

int Rseed = time(NULL) ;

void randSeed(int seed){

randNum = seed;

}

/\*\*线性同余： Tn = (A\*Tn-1+B) mod C 易知Tn 为 0~C 之间的数

\* @author Swking

\* @method pseudoRand

\* @parame

\* @return randNum 伪随机数

\*/

double pseudoRand(){

randNum = ((randNum\*9301)%49853 + 49297)%49999 ;

return randNum;

}

/\*\*

\* 49919 49921 49927 49937 49939 49943 49957 49991 49993 49999

\* 48857 48859 48869 48871 48883 48889 48907 48947 48953 48973

\* 47143 47147 47149 47161 47189 47207 47221 47237 47251 4726

\*

\*/

int main()

{

fstream fin, fout;

fin.open("randin.txt",ios::in|ios::out);

fout.open("randout.txt",ios::out);

srand(time(NULL));

int examNum = 100;

while(examNum--){

fin << 1000+rand()%50 << endl;

fin << 1 << ' ' << 20+rand()%20 <<endl;

}

fin.close();

fin.open("randin.txt",ios::in|ios::out);

randSeed(Rseed);

int couter = 0;

while(!fin.eof()){

int n=0;

int a,b;

int num;

fin >> n;

int N=n;

if(n!=0){

fin >> a >> b;

int t[b+1];

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

t[i]=0;

}

fout << "-- "<<couter<<" --" << "a:"<<a<<" --"<<"b:"<<b<<" ---"<<endl;

couter++;

while(n--){

num = a+(int)pseudoRand()%(b-a);

fout << num << ' ';

t[num] = t[num] + 1;

}

for(int i=a; i<b; i++){

float re = t[i]\*100\*1.00/N;

fout<< i<< ":"<< re <<"%"<<endl;

}

}

fout << endl <<endl;

}

fout << endl;

cout << "Completed!" << endl;

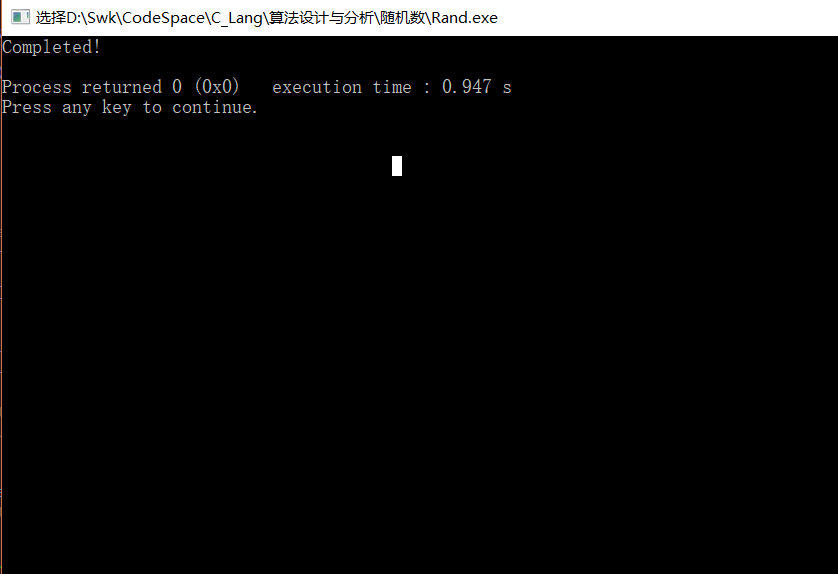
fout.close();

fin.close();

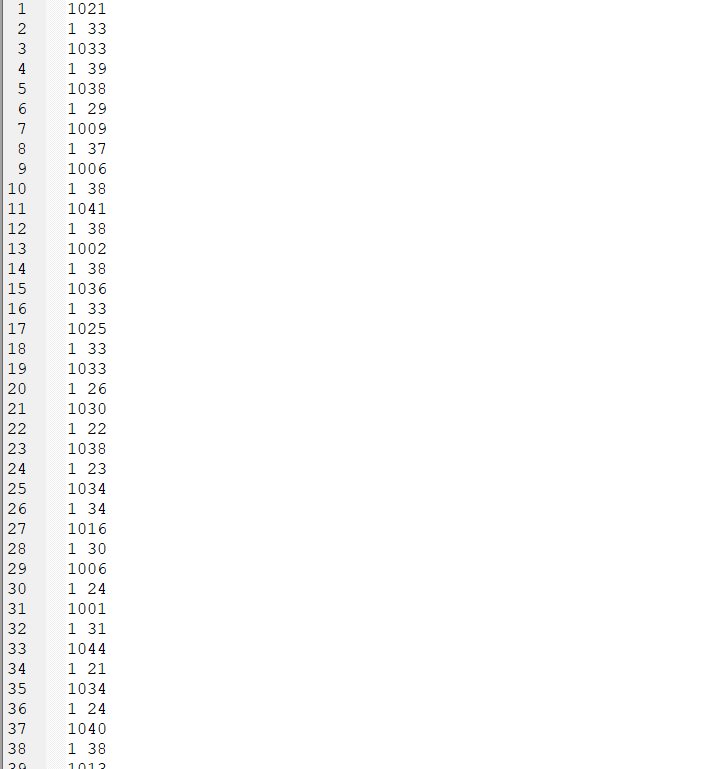
return 0;

}

1. **程序运行结果**
2. 控制台输出信息



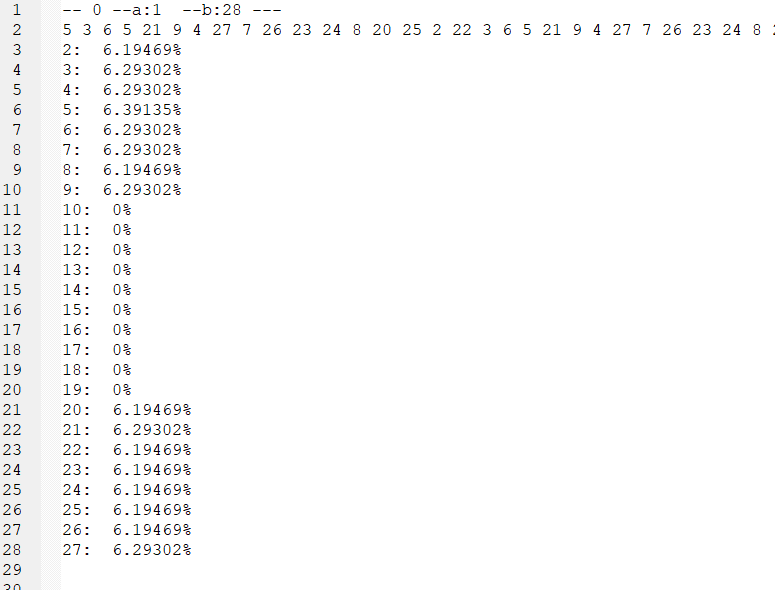
1. 数据文件randin.txt中，第一行是要产生随机数的个数，第二行两个数分别为a和b。



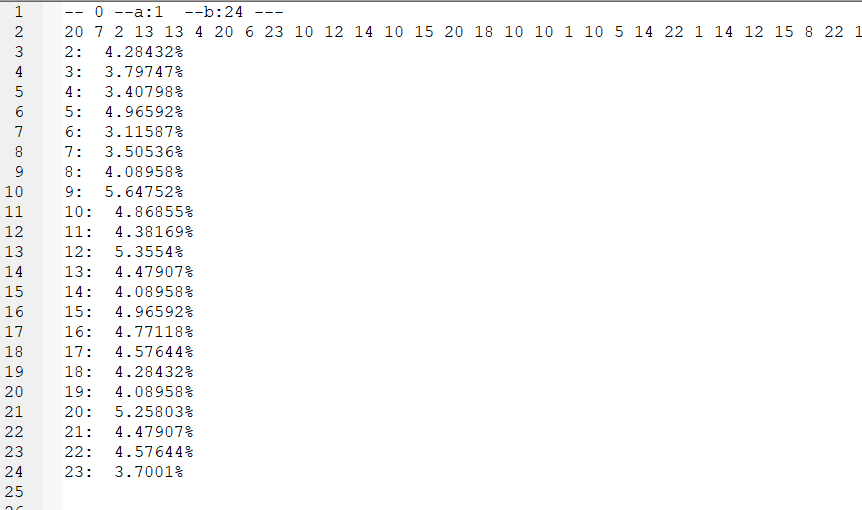
1. 运行程序将结果输出到randout.txt文件中，如图：第一行为样例信息，第二行为产生的随机数，下面为产生的随机数出现的概率，用于分析程序效率。



1. **实验结果分析**
2. 当线性同余Tn = (A\*Tn-1+B) mod C，A、B、C三个参数选得很小时，就会出现部分数字不出现，且重复率很高，但是若数字出现，则概率相近



2、若3个参数取得很大时结果如图，数字产生比较均匀，可知选择参数的大小对伪随机数的产生有很大影响。



1. 有时候会由于种子过大，而参数C不够大导致随机数产生的不够均匀，随意算法改进成为Tn = (((A\*Tn-1) mod D) + B) mod C,增加一个参数D，使得种子序列的初始值减小，这样使得产生的伪随机数能够被整除的概率减小。
2. **结论**

1、 线性同余Tn = (A\*Tn-1+B) mod C，通过一个同余的递推式来产生伪随机数，关键在于A、B、C三个参数和序列的种子的挑选。

2、其中A、B、C互质就可以获得概率较平均的伪随机数，但是最终影响伪随机数范围的参数是C，决定产生随机数的大小及概率，所以挑选一个较大质数可以达到理想的情况。

3、 还有一个是关键因素是种子的挑选，好的种子直接影响到随机数的正常生成。例如可以用时间作为种子，或者未赋值的静态变量地址作为种子等等。