Міністерство освіти і науки України

Національний Технічний Університет

«Харківський Політехнічний Інститут»

Кафедра «Стратегічного Керування»

Лабораторна робота № 5

«КОМБІНАТОРНІ АЛГОРИТМИ»

Перевірила: Мошко Є. О.

Виконав:

Харків, 2018

Лабораторна робота № 5

Тема: Комбінаторні Алгоритми

Завдання 1

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.

(2 варіант)

* Лінійний конгруентний метод.
* Гістограма розподілу елементів послідовності.
* Тест на послідовність однакових бітів.

Хід роботи

При N=50; m=50; x=1; a=11; c=3;

|  |
| --- |
| #include <iostream>  #include <fstream>  #include <bitset>  #include <string>  #include <cmath>  using namespace std;  int lcg(int m = 0, int a = 0, int c = 0, int x = 0){  static int M = m;  static int A = a;  static int C = c;  static int X = x;  X = (A\*X+C) % M;  return X;  }  void distribution(int\* arr\_ran, int N, int m){  ofstream out("data.txt");  int size = 10; // columns of histogram  int\* distr = new int[size];  distr[0]=m/(size);  for (int i=1; i<size; i++){  distr[i]=distr[i-1]+m/(size);  }  int\* tmp = new int[size];  for (int i = 0;i<size; i++) tmp[i]=0;  for (int i=0; i<N; i++){  for (int j=0; j<size; j++){  if (arr\_ran[i]<distr[j]){  tmp[j]++;  break;  }  }  }  int middle=m/size/2;  for (int i=0; i<size; i++){  out << distr[i]-middle << "\t" << tmp[i] << endl;  }  }  void distribution2(int\* arr\_ran, int N){  ofstream out("data2.txt");  for (int i=0;i<N; i+=2){  out << arr\_ran[i] << "\t" << arr\_ran[i+1] << endl;  }  }  void test(int\* arr\_ran, int N){  string str;  int ones=0;  for (int i=0; i<N; i++){  bitset<8> b{arr\_ran[i]};  str += b.to\_string();  ones += b.count();  cout << b.to\_string() << endl;  }  cout << endl << "1) Ones: " << ones << endl;  float pi = (float)ones/N/8;  cout << "2) Pi = Ones/n: " << pi << endl;  if (abs(pi-0.5)<2/sqrt((float)(N\*8))){  cout << "3) |pi-0.5|<2/sqrt(n): " << abs(pi-0.5) << " < " << 2/sqrt((float)(N\*8)) << endl;  int v=0;  for (int i=0; i<N\*8-1; i++){  if (str[i]!=str[i+1]) v++;  }  v++;  cout << "4) Total leaps v: " << v << endl;  double P = erfc(abs(v-2\*N\*8\*pi\*(1-pi))/(2\*sqrt(2\*N\*8)\*pi\*(1-pi)));  if (P>=0.01){  cout << "5) P: " << P << " >= 0.01" << endl;  }  else  {  cout << "5) P: Failed! " << P << " < 0.01" << endl;  }  }  else{  cout << "3) |pi-0.5|<2/sqrt(n): Failed! " << abs(pi-0.5) << " >= " << 2/sqrt((float)(N\*8)) << endl;  return;  }  }  int main() {  int N, a, c, x;  int m;    cout << "Linear Congruent Generator" << endl;  cout << "Enter number of digits N, modulus m, multiplier a, increment c, start value X: " << endl;  //cin >> N >> m >> a >> c >> x;  N=10;  m=100;  x=1;  a=21;  c=3;    int\* arr\_ran = new int[N];  lcg(m,a,c,x);  for (int i=0; i<N; i++){  int get\_ran = lcg();  cout << get\_ran << " ";  arr\_ran[i]=get\_ran;  }  cout << endl << endl;  distribution(arr\_ran, N, m);  test(arr\_ran, N);  system("pause");  } |

Приклад виконання програми:

|  |
| --- |
| Linear Congruent Generator  Enter number of digits N, modulus m, multiplier a, increment c, start value X:  7 30 33 16 29 22 45 48 31 44 37 10 13 46 9 2 25 28 11 24 17 40 43 26 39 32 5 8 41 4 47 20 23 6 19 12 35 38 21 34 27 0 3 36 49 42 15 18 1 14  00000111  00011110  00100001  00010000  00011101  00010110  00101101  00110000  00011111  00101100  00100101  00001010  00001101  00101110  00001001  00000010  00011001  00011100  00001011  00011000  00010001  00101000  00101011  00011010  00100111  00100000  00000101  00001000  00101001  00000100  00101111  00010100  00010111  00000110  00010011  00001100  00100011  00100110  00010101  00100010  00011011  00000000  00000011  00100100  00110001  00101010  00001111  00010010  00000001  00001110  1) Ones: 133  2) Pi = Ones/n: 0.3325  3) |pi-0.5|<2/sqrt(n): Failed! 0.1675 >= 0.1  Press any key to continue . . . |

Запис до файлу data.txt

|  |
| --- |
| 3 5  8 5  13 5  18 5  23 5  28 5  33 5  38 5  43 5  48 5 |

Приклад гістограми в Excel

|  |
| --- |
|  |

Висновок

Я ознайомився з лінійним конгруентним методом генераціі псевдовипадкових значень. Послідовність отриманих чисел не володіє криптографічною стійкістю. Потребує правильне налаштування коефіцієнтів a – множника, m – модуля (період), c – приріст. Тест на послідовність однакових бітів полягає в підрахунку повного числа рядів у вихідній послідовності. У цій лабі тест не було пройдено.