



**IIT MADRAS**  
Indian Institute of Technology Madras

**M.Tech Computer Science Information security.**

**Cryptography Basic.(CS6530)**

**Assignment-2. (Question 8.5 Page number-281 programming )**

**Roll Number - CS21M515**

**Name – Ayub Shaikh.**

**Batch Year-2021**

**(Question:- 8.5 - Page number-281 programming )**

- 8.5 In any use of pseudorandom numbers, whether for encryption, simulation, or statistical design, it is dangerous to trust blindly the random number generator that happens to be available in your computer's system library. [PARK88] found that many contemporary textbooks and programming packages make use of flawed algorithms for pseudorandom number generation. This exercise will enable you to test your system.

The test is based on a theorem attributed to Ernesto Cesaro (see [KNUT98] for a proof), which states the following: Given two randomly chosen integers,  $x$  and  $y$ , the probability that  $\gcd(x, y) = 1$  is  $6/\pi^2$ . Use this theorem in a program to determine statistically the value of  $\pi$ . The main program should call three subprograms: the random number generator from the system library to generate the random integers; a subprogram to calculate the greatest common divisor of two integers using Euclid's Algorithm; and a subprogram that calculates square roots. If these latter two programs are not available, you will have to write them as well. The main program should loop through a large number of random numbers to give an estimate of the aforementioned probability. From this, it is a simple matter to solve for your estimate of  $\pi$ .

If the result is close to 3.14, congratulations! If not, then the result is probably low, usually a value of around 2.7. Why would such an inferior result be obtained?

### Solution and Answer:-

I Have did the above example by using the C++ program.

And .exe is shared in the google drive and also the program is shared.

Below are the screen shots of the C++ programm executed .

So in this belwo screen shot I have generated 20 random numbers and then calculated their GCD and propbaility of that and then value of Pi.

So in below screen shot as we can see that I am getting the approximate value of pi is = **3.16228**

So it means the random numbers are generated properly and program is also correct.

Select "C:\Users\DELL\Desktop\IIT Mtech\Crypto\newProg\PseudorandomNumbersVerification\bin\Debug\Pseudorandom

```
-----Assignment-1 Submitted by Roll number: CS21M515 -----
Enter the number so that how many of random numbers you want
20
Random numbers are:
230
526
165
538
712
853
805
949
801
1
390
340
79
430
876
720
448
158
19
353
GCD(230, 526) = 2
GCD(165, 538) = 1
GCD(712, 853) = 1
GCD(805, 949) = 1
GCD(801, 1) = 1
GCD(390, 340) = 10
GCD(79, 430) = 1
GCD(876, 720) = 12
GCD(448, 158) = 2
GCD(19, 353) = 1
Total number (a,b) groups are: 10
Euclidean Algorithm GCD(a,b)==1 equal to one, count are: 6
So the Probability of GCD(a,b) equal to 1 is : (6/10) = 0.6
So the final estimated pi value is : 3.16228
press any number to exit:
```

So in this below screen shot I have generated 10000 random numbers and then calculated their GCD and probability of that and then value of Pi.

So in below screen shot as we can see that I am getting the approximate value of pi is = **3.14399**

So it means the random numbers are generated properly and program is also correct.

Select "C:\Users\DELL\Desktop\IIT Mtech\Crypto\newProg\PseudorandomNumbersVerification\bin\Debug\PseudorandomNumbersVerification.exe"

```
GCD(133, 523) = 1
GCD(909, 932) = 1
GCD(236, 245) = 1
GCD(166, 270) = 2
GCD(681, 556) = 1
GCD(596, 816) = 4
GCD(459, 510) = 51
GCD(671, 94) = 1
GCD(182, 854) = 14
GCD(373, 635) = 1
GCD(737, 120) = 1
GCD(553, 14) = 7
GCD(648, 678) = 6
GCD(872, 724) = 4
GCD(314, 146) = 2
GCD(568, 661) = 1
GCD(510, 53) = 1
GCD(754, 572) = 26
GCD(521, 431) = 1
GCD(777, 985) = 1
GCD(427, 791) = 7
GCD(491, 489) = 1
GCD(12, 854) = 2
GCD(616, 609) = 7
GCD(513, 1000) = 1
GCD(857, 214) = 1
GCD(419, 539) = 1
GCD(658, 833) = 7
GCD(98, 130) = 2
GCD(971, 913) = 1
GCD(21, 148) = 1
GCD(179, 114) = 1
GCD(685, 305) = 5
GCD(374, 444) = 2
GCD(355, 748) = 1
GCD(418, 327) = 1
GCD(607, 167) = 1
GCD(804, 478) = 2
GCD(75, 996) = 3
Total number (a,b) groups are: 5000
Euclidean Algorithm GCD(a,b)==1 equal to one, count are: 3035
So the Probability of GCD(a,b) equal to 1 is : (3035/5000) = 0.607
So the final estimated pi value is : 3.14399
press any number to exit: _
```

So in this I have generated 5000 random numbers and then calculated their GCD and propbaility of that and then value of Pi.

So in below screen shot as we can see that I am getting the approximate value of pi is = **3.13934**

So it means the random numbers are generated properly and program is also correct.

Select "C:\Users\DELL\Desktop\IIT Mtech\Crypto\newProg\PseudorandomNumbersVerification\bin\Debug\PseudorandomNumbersVerification.exe"

```
GCD(999, 853) = 1
GCD(397, 224) = 1
GCD(888, 289) = 1
GCD(369, 36) = 9
GCD(795, 807) = 3
GCD(357, 218) = 1
GCD(745, 980) = 5
GCD(10, 405) = 5
GCD(936, 119) = 1
GCD(150, 164) = 2
GCD(645, 874) = 1
GCD(299, 369) = 1
GCD(466, 14) = 2
GCD(583, 879) = 1
GCD(24, 750) = 6
GCD(246, 186) = 6
GCD(190, 823) = 1
GCD(478, 242) = 2
GCD(547, 726) = 1
GCD(393, 776) = 1
GCD(361, 498) = 1
GCD(344, 844) = 4
GCD(443, 20) = 1
GCD(229, 990) = 1
GCD(128, 997) = 1
GCD(234, 59) = 1
GCD(8, 959) = 1
GCD(560, 842) = 2
GCD(254, 70) = 2
GCD(75, 496) = 1
GCD(687, 696) = 3
Total number (a,b) groups are: 2500
Euclidean Algorithm GCD(a,b)==1 equal to one, count are: 1522
So the Probability of GCD(a,b) equal to 1 is : (1522/2500) = 0.6088
So the final estimated pi value is : 3.13934
press any number to exit:
```

So in this I have generated 10000 random numbers and then calculated their GCD and probability of that and then value of Pi.

So in below screen shot as we can see that I am getting the approximate value of pi is = **3.13266**

So it means the random numbers are generated properly and program is also correct.

Select "C:\Users\DELL\Desktop\IIT Mtech\Crypto\newProg\PseudorandomNumbersVerification\bin\Debug\PseudorandomNumbersVerification.exe"

```
GCD(10642, 2180) = 2
GCD(6208, 8971) = 1
GCD(8307, 10429) = 1
GCD(7699, 1815) = 1
GCD(8458, 8986) = 2
GCD(815, 6410) = 5
GCD(438, 10456) = 2
GCD(1638, 9216) = 18
GCD(5049, 7692) = 3
GCD(5093, 7621) = 1
GCD(8012, 3838) = 2
GCD(1441, 7651) = 1
GCD(3709, 6458) = 1
GCD(5949, 6701) = 1
GCD(8957, 10879) = 1
GCD(5745, 8878) = 1
GCD(6561, 9824) = 1
GCD(10400, 6822) = 2
GCD(7224, 10688) = 8
GCD(3943, 10830) = 1
GCD(10047, 406) = 1
GCD(2176, 8807) = 1
GCD(2699, 10354) = 1
GCD(381, 2743) = 1
GCD(3810, 0) = 3810
Total number (a,b) groups are: 5000
Euclidean Algorithm GCD(a,b)==1 equal to one, count are: 3057
So the Probability of GCD(a,b) equal to 1 is : (3057/5000) = 0.6114
So the final estimated pi value is : 3.13266
press any number to exit:
```

"C:\Users\DELL\Desktop\IIT Mtech\Crypto\newProg\PseudorandomNumbersVerification\bin\Debug\PseudorandomNumbersVerification.exe"

```
GCD(10642, 2180) = 2
GCD(6208, 8971) = 1
GCD(8307, 10429) = 1
GCD(7699, 1815) = 1
GCD(8458, 8986) = 2
GCD(815, 6410) = 5
GCD(438, 10456) = 2
GCD(1638, 9216) = 18
GCD(5049, 7692) = 3
GCD(5093, 7621) = 1
GCD(8012, 3838) = 2
GCD(1441, 7651) = 1
GCD(3709, 6458) = 1
GCD(5949, 6701) = 1
GCD(8957, 10879) = 1
GCD(5745, 8878) = 1
GCD(6561, 9824) = 1
GCD(10400, 6822) = 2
GCD(7224, 10688) = 8
GCD(3943, 10830) = 1
GCD(10047, 406) = 1
GCD(2176, 8807) = 1
GCD(2699, 10354) = 1
GCD(381, 2743) = 1
GCD(3810, 0) = 3810
Total number (a,b) groups are: 5000
Euclidean Algorithm GCD(a,b)=1 equal to one, count are: 3057
So the Probability of GCD(a,b) equal to 1 is : (3057/5000) = 0.6114
So the final estimated pi value is : 3.13266
press any number to exit:
```

"C:\Users\DELL\Desktop\IIT Mtech\Crypto\newProg\PseudorandomNumbersVerification\bin\Debug\PseudorandomNumbersVerification.exe"

```
GCD(319, 931) = 1
GCD(218, 973) = 1
GCD(744, 725) = 1
GCD(553, 663) = 1
GCD(60, 769) = 1
GCD(318, 688) = 2
GCD(791, 274) = 1
GCD(751, 387) = 1
GCD(46, 213) = 1
GCD(656, 56) = 8
GCD(464, 587) = 1
GCD(909, 347) = 1
GCD(770, 680) = 10
GCD(797, 54) = 1
GCD(163, 626) = 1
GCD(648, 544) = 8
GCD(330, 155) = 5
GCD(602, 694) = 2
GCD(356, 171) = 1
GCD(768, 217) = 1
GCD(683, 463) = 1
GCD(602, 80) = 2
GCD(643, 934) = 1
GCD(754, 823) = 1
GCD(447, 310) = 1
GCD(693, 248) = 1
GCD(219, 475) = 1
GCD(39, 897) = 39
GCD(27, 976) = 1
GCD(300, 575) = 25
GCD(364, 961) = 1
GCD(468, 850) = 2
GCD(233, 479) = 1
GCD(502, 40) = 2
GCD(794, 277) = 1
GCD(608, 158) = 2
GCD(368, 532) = 4
Total number (a,b) groups are: 100
Euclidean Algorithm GCD(a,b)=1 equal to one, count are: 60
So the Probability of GCD(a,b) equal to 1 is : (60/100) = 0.6
So the final estimated pi value is : 3.16228
press any number to exit:
```

## Program in C++ language

```
#include<iostream>
```

```
#include<cstdlib>
```

```
#include <ctime>
```

```
#include <cmath>
```

```
//-----Assignment-2 Submitted by Roll number: CS21M515 -
```

```
using namespace std;
```



```
// GCD
```

```
// Function to return
```

```
// gcd of a and b
```

```
int gcd(int a, int b)
```

```
{
```

```
    if (a == 0)
```

```
        return b;
```

```
    return gcd(b % a, a);
```

```
}
```

```
/// end
```

```
int main(){
```

```
    int n;
```

```
    int m;
```

```
    int t;
```

```
    int c;
```

```
    cout << "-----" << endl;
```

```
    cout << "-----Assignment-2 Submitted by Roll number: CS21M515 -----" << endl;
```

```
    cout << "-----" << endl;
```

```
    cout << "Enter the number so that how many of random numbers you want" << endl;
```

```
    cin >> n;
```

```
    cout << "Random numbers are:" << endl;
```

```

int randomNumArray[n];

float prob;

int range = 1000;

if(n > range){
    range = n + range;
}

// Providing a seed value
srand((unsigned) time(NULL));

m = n+1;

// Loop to get 5 random numbers

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

    // Retrieve a random number between 100 and 200

    // Offset = 1

    // Range = 1000

    int random = 1 + (rand() % range);

    randomNumArray[i] = random;

    // Print the random number

    cout<<random<<endl;

}

float euclideanGCDCCount = 0.0;

```

```

for(int i=0; i < n; i++){
    // cout << randomNumArray[i] << " " << endl;

    int a = randomNumArray[i];

    int b = randomNumArray[++i];

    int g = gcd(a, b);
    cout << "GCD(" << a << ", " << b << ") = " << g << endl;

    if(g == 1){
        euclideanGCDCount++;
    }
}

// cout << "i(" << i << endl;

    float numGroups = n/2;

    cout << " Total number (a,b) groups are: " << numGroups << endl;

    cout << " Euclidean Algorithm GCD(a,b)==1 equal to one, count are: " <<
euclideanGCDCount << endl;

    prob = euclideanGCDCount / numGroups;

    cout << "So the Probability of GCD(a,b) equal to 1 is : (" << euclideanGCDCount << "/"
<< numGroups << ") = " << prob << endl;

```

```
float cal = 6/prob;
```

```
float pi = sqrt(cal);
```

```
cout << " So the final estimated pi value is : " << pi << endl;
```

```
int ex;
```

```
cout << "press any number to exit: ";
```

```
cin >> ex;
```

```
if(ex>0){
```

```
    return 0;
```

```
}
```

```
//return 1;
```

```
}
```

```
"C:\Users\DELL\Desktop\UIT Mtech\Crypto\newProg\PseudorandomNumbersVerification\bin\Debug\PseudorandomNumbersVerification.exe"
GCD(1757, 766) = 1
GCD(943, 926) = 1
GCD(1116, 1940) = 4
GCD(401, 635) = 1
GCD(1438, 1987) = 1
GCD(822, 870) = 6
GCD(1307, 1907) = 1
GCD(116, 1859) = 1
GCD(109, 372) = 1
GCD(1885, 324) = 1
GCD(44, 573) = 1
GCD(667, 719) = 1
GCD(1352, 707) = 1
GCD(1328, 1057) = 1
GCD(1779, 1756) = 1
GCD(1383, 1491) = 3
GCD(1124, 548) = 4
GCD(1119, 156) = 3
GCD(724, 703) = 1
GCD(249, 264) = 3
GCD(1491, 1152) = 3
GCD(1072, 824) = 8
GCD(1849, 1258) = 1
GCD(105, 1737) = 3
GCD(1776, 117) = 3
GCD(1691, 1801) = 1
GCD(896, 580) = 4
GCD(160, 1127) = 1
GCD(235, 804) = 1
GCD(1132, 59) = 1
GCD(491, 967) = 1
GCD(78, 1782) = 6
GCD(994, 706) = 2
GCD(1627, 1643) = 1
GCD(1886, 771) = 1
GCD(1988, 814) = 2
GCD(398, 1528) = 2
Total number (a,b) groups are: 1000
Euclidean Algorithm GCD(a,b)=1 equal to one, count are: 615
So the Probability of GCD(a,b) equal to 1 is : (615/1000) = 0.615
So the final estimated pi value is : 3.12348
press any number to exit: _
```

THANK You Sir .... 😊😊😊