Kodas:

```python
from Golomb import getTests, printCheck
import random


def linearReg(c):
    X = []
    for i in range(len(c)):
        X.append(int(round(random.random(), 0)))

    n = len(X)
    m = pow(2, n) - 1
    c = c[::-1]
    string = ""

    for i in range(m):
        string += str(X[n-1])
        temp = 0
        for j in range(n):
            temp += int(X[j])*int(c[j])

        j = n-1
        while j > 0:
            X[j] = X[j-1]
            j -= 1

        X[0] = str(temp % 2)

    return string
```

Išvedimas:

```python
string = linearReg("1000001")
print (string)
printCheck(string, 0.05)

string = linearReg("1000111")
print (string)
printCheck(string, 0.05)

string = linearReg("1100101")
print (string)
printCheck(string, 0.05)

string = linearReg("10010101")
print (string)
printCheck(string, 0.05)

string = linearReg("10110001")
print (string)
printCheck(string, 0.05)

string = linearReg("11100001")
print (string)
printCheck(string, 0.05)
```

Išbandžiau algoritmą su 6 skirtingais C vektoriais, gautais iš 7 ir 8 laipsnio primityviųjų daugianarių ir patikrinau, kad jie tikrai išveda maksimalaus ilgio m bitų seką išvesdamas 10 daugiau simbolių nei m ir patikrindamas kad pirmi 10 simbolių sutampa tiktai su paskutiniais 10 simbolių.

Rezultatai:

```
[Running] python -u "c:\Users\Marius\Desktop\7 semestras\informacinės saugos pagrindai\uzduotys\Golomb\5LD.py"
11011010110110010010001111000010111110010101110011010001001111000101000011000001000000111111101010010011011101100101100011011
{'T1': 0.007874015748031496, 'T2': 0.023872015997994822, 'T4': 0.501937984496124, 'T5': 0.25, 'T1check': 3.841458820694124, 'T2check': 5
T1 - Pass
T2 - Pass
T4 - Pass
T5 - Pass
10111000010011000001010101101001001010011110010001101010000111111101110111011010001011001011111100010000001100110110001111001110
{'T1': 0.007874015748031496, 'T2': 0.023872015997994822, 'T4': 0.001937984496124031, 'T5': 1.0, 'T1check': 3.841458820694124, 'T2check':
T1 - Pass
T2 - Pass
T4 - Pass
T5 - Pass
10100010111000111101110110101111111010011000011010000101001011011001010101100010000010011111001110010010001100110111110000001110
{'T1': 0.007874015748031496, 'T2': 0.023872015997994822, 'T4': 0.001937984496124031, 'T5': 0.75, 'T1check': 3.841458820694124, 'T2check'
T1 - Pass
T2 - Pass
T4 - Pass
T5 - Pass
00101001100110111101010101100011111111010010000000111011000000100110100000011010111000010111100100011100010110010010011101011011010011110
{'T1': 0.00392156862745098, 'T2': 0.011826462868611998, 'T4': 0.001217860685130083 2, 'T5': 1.414213562373095, 'T1check': 3.8414588206941
T1 - Pass
T2 - Pass
T4 - Pass
T5 - Pass
01100001010010111101001111111101010000111001111000011011001101000111101110001110100000001111100010010100010001011010101110010001100100100
{'T1': 0.00392156862745098, 'T2': 0.011826462868611998, 'T4': 0.001217860685130083 2, 'T5': 0.7071067811865475, 'T1check': 3.841458820694
T1 - Pass
T2 - Pass
T4 - Pass
T5 - Pass
00101001011010111110001011001000010100010010010101010011001011010001110110011111011010000001000001011110111011110000100111010110001111
{'T1': 0.00392156862745098, 'T2': 0.011826462868611998, 'T4': 0.001217860685130083 2, 'T5': 0.8838834764831843, 'T1check': 3.841458820694
T1 - Pass
T2 - Pass
T4 - Pass
T5 - Pass
```

Paskaičiavau testines reikšmes Ti ir reikšmes pagal kurias reikia patikrinti ar seka praeina testą – Ticheck (Ti turi būti mažiau už Ticheck, skaičiuojant T5 d = n/2). Jei seka praeina testą išvedama „Pass“, jei ne išvedama „Fail“

Kai C = [1,0,0,0,0,1]:

Seka:
11011010110110010010001111000010111110010101110011010001001111000101000011000001000000111111101010010011001110111010010110001111011

T1 = 0.007874015748031496	T1check = 3.841458820694124	Pass
T2 = 0.023872015997994822	T2check = 5.991464547107979	Pass
T4 = 0.501937984496124	T4check = 9.487729036781154	Pass
T5 = 0.25	T5check = 1.959963984540054	Pass

Kai C = [1,0,0,0,1,1,1]:

Seka:
10111000010011000001010101101001001010011110010001101010000111111101110110111
01000101100101111100010000001100110110001110011110

Kai C = [1,1,0,0,1,0,1]:

Seka:
10100010111000111101110110101111111010011000011010000101001011011001010101100
10000010011111001110010010001100110111100000011110

Kai C = [1,0,0,1,0,1,0,1]:

Seka:
00101001100110111101010101100011111111010010000000111011000000100110100000110
10111000010111100100011100010110010010011101011011010011110110111010001101100
11001011010010111011111011100110000110010101000101011111100111110000010100001
00001111000110001 0001

Kai C = [1,0,1,1,0,0,0,1]:

Seka:
01100001010010111101001111111101010000111001111000011011001101000111101110001
10100000001111100010010100010001011010101110010001100100100111000000100001001
10111001100111011111100101100101011011010010000011000110101100010101010011000
00101110101111101101 11

Kai C = [1,1,1,0,0,0,0,1]:

Seka:
00101001011010111110001011001000010100010010010101010011001011101000111011001
11101101000000100000101111011101111000010011101011000111100110110111111110101
01110011100100110101000011011100000110000111000110011000100011010011110100100
01010110100000000111 11