Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Фізико-технічний інститут

**ЛАБОРАТОРНА РОБОТА №3**

**з дисципліни**

**«Криптографія»**

**на тему: «Побудова регістрів зсуву з лінійним зворотним зв'язком та дослідження їх властивостей»**

**Варіант 2**

Виконали:
студенти 3 курсу ФТІ
групи ФБ-73
Маковецький А.О.
Бадарак О.А.

Перевірили:
Чорний О.
Савчук М. М.
Завадська Л. О.

**Мета роботи:** Ознайомлення з принципами побудови регістрів зсуву з лінійним зворотним зв'язком; практичне освоєння їх програмної реалізації; дослідження властивостей лінійних рекурентних послідовностей та їх залежності від властивостей характеристичного полінома регістра.

## Порядок виконання роботи

0. Уважно прочитати методичні вказівки до виконання комп'ютерного практикуму.
1. Вибрати свій варіант завдання згідно зі списком. Варіанти завдань містяться у файлі Crypto_CP4 LFSR_Var.
2. За даними характеристичними многочленами p1(x), p2(x) скласти лінійні рекурентні співвідношення для ЛРЗ, що задаються цими характеристичними многочленами.
3. Написати програми роботи кожного з ЛРЗ L1 , L2 .
4. За допомогою цих програм згенерувати імпульсні функції для кожного з ЛРЗ і підрахувати їх періоди.
5. За отриманими результатами зробити висновки щодо влавстивостей кожного з характеристичних многочленів p1(x), p2(x): многочлен примітивний над F2 ; не примітивний, але може бути незвідним; звідний.
6. Для кожної з двох імпульсних функцій обчислити розподіл *k*-грам на періоді, $k \le n_i$, де $n_i$ - степінь полінома $f_i(x)$, $i$=1,2 а також значення функції автокореляції $A(d)$ для $0 \le d \le 10$. За результатами зробити висновки.

| | |
|---|---|
| 2 | $P_1(X) = X^{22} + X^{17} + X^{16} + X^{15} + X^{14} + X^{12} + X^{11} + X^7 + X^6 + X^5 + X^3 + X + 1$<br>$P_2(X) = X^{21} + X^{18} + X^{14} + X^9 + X^8 + X^2 + 1$ |

**P1**:
Period: 1398101
The polynom is reducible
autocorrelation for d=0: 0
autocorrelation for d=1: 698368
autocorrelation for d=2: 699392
autocorrelation for d=3: 699392
autocorrelation for d=4: 698368
autocorrelation for d=5: 699392
autocorrelation for d=6: 698368
autocorrelation for d=7: 699392
autocorrelation for d=8: 699392
autocorrelation for d=9: 698368
autocorrelation for d=10: 698368

**Кількість n-грам:**

Ngram length: 1
{'0': 698709, '1': 699392}

Ngram length: 2
{'00': 349525, '01': 349184, '10': 349183, '11': 350208}

Ngram length: 3
{'000': 174933, '001': 174592, '010': 174080, '100': 174591, '011': 175104, '111': 175104, '110': 175103, '101': 174592}

Ngram length: 4
{'0000': 87381, '0001': 87552, '0010': 87040, '0100': 87040, '1000': 87551, '0011': 87552, '0111': 87552, '1111': 87552, '1110': 87551, '1100': 87551, '1001': 87040, '0110': 87552, '1101': 87552, '1010': 87040, '0101': 87040, '1011': 87552}

Ngram length: 5
{'00000': 43669, '00001': 43712, '00010': 43584, '00100': 43456, '01000': 43712, '10000': 43711, '00011': 43968, '00111': 43584, '01111': 43712, '11110': 43712, '11100': 43711, '11001': 43712, '10011': 43584, '00110': 43968, '01101': 43712, '11010': 43584, '10101': 43456, '01011': 43584, '10110': 43584, '11011': 43968, '01010': 43456, '10100': 43584, '10001': 43840, '11111': 43840, '11000': 43839, '01110': 43839, '11101': 43840, '10111': 43968, '01001': 43328, '10010': 43456, '00101': 43584, '01100': 43840}

**P2:**
Period: 2097151
The polynom is primitive
autocorrelation for d=0: 0
autocorrelation for d=1: 1048576
autocorrelation for d=2: 1048576
autocorrelation for d=3: 1048576
autocorrelation for d=4: 1048576
autocorrelation for d=5: 1048576
autocorrelation for d=6: 1048576
autocorrelation for d=7: 1048576
autocorrelation for d=8: 1048576
autocorrelation for d=9: 1048576
autocorrelation for d=10: 1048576

**Кількість n-грам:**

Ngram length: 1
{'0': 1048575, '1': 1048576}

Ngram length: 2
{'00': 524287, '01': 524288, '10': 524287, '11': 524288}

Ngram length: 3
{'000': 262143, '001': 262144, '010': 262143, '100': 262143, '011': 262144, '110': 262144, '101': 262144, '111': 262144}

Ngram length: 4
{'0000': 131071, '0001': 131072, '0010': 131072, '0100': 131071, '1001': 131072, '0011': 131072, '0110': 131072, '1101': 131072, '1010': 131071, '1000': 131071, '0101': 131072, '0111': 131072, '1110': 131072, '1011': 131072, '1100': 131072, '1111': 131072}

Ngram length: 5
{'00000': 65535, '00001': 65536, '00010': 65536, '00100': 65536, '01001': 65536, '10010': 65536, '10011': 65536, '00110': 65536, '01101': 65536, '11010': 65536, '10100': 65535, '01000': 65535, '10000': 65535, '00011': 65536, '10101': 65536, '01010': 65535, '00111': 65536, '01110': 65536, '11101': 65536, '00101': 65536, '01011': 65536, '10110': 65536, '01100': 65536, '11001': 65536,

'01111': 65536, '11111': 65536, '11110': 65536, '11011': 65536, '10001': 65536, '11000': 65536, '10111': 65536, '11100': 65536}

**Код програми:**

```
# var2
from collections import deque

p1 = (1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0)
p2 = (1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0)

def lfsr(poly, outputfile):
	register = deque()
	period = 0
	f = open(outputfile, 'w', encoding='utf-8')

	# Initialize register state (impulse function)
	for i in range(len(poly) - 1):
		register.append(0)
	register.append(1)

	start_state = register.copy()

	while True:
		temp = register[0] * poly[0]
		for i in range(1, len(register)):
			temp = temp ^ (register[i] * poly[i])
		f.write(str(register.popleft()))
		period += 1
		register.append(temp)
		if register == start_state:
			f.close()
			print("LFSR result written to " + outputfile)

			return period

def ngram_count(text, ngram_len): # How should i count it???
	ngram_count = {}
	for i in range(len(text) - ngram_len + 1):
		ngram = text[i : i + ngram_len]
		try:
			ngram_count[ngram] += 1
		except:
			ngram_count[ngram] = 1

	return ngram_count

def autocorrelation(arr, period, d):
	sum = 0
	for i in range(period):
		sum += (int(arr[i]) + int(arr[(i + d) % period])) % 2
	return sum

def ngram_count_task(text, max_len, filename):
	with open(filename, 'w', encoding='utf-8') as f:
		for length in range(1, max_len + 1):
			f.write('Ngram length: ' + str(length) + '\n')
			f.write('Ngrams:\n' + str(ngram_count(text, length)) + '\n\n')
	print('Ngrams count written to ' + filename)
```

```
def autocorrelation_task(arr, period):
	res = {}
	for d in range(11):
		res[d] = autocorrelation(arr, period, d)
		print('autocorrelation for d=' + str(d) + ':', autocorrelation(arr,
period, d))
	return res

def import_data(filename):
	with open(filename, 'r', encoding='utf-8') as f:
		return f.read()


def main():

	print('\n==== POLYNOM 1 ====\n')
	period1 = lfsr(p1, 'LFSR_result1.txt')

	print('Period:', period1)

	data1 = import_data('LFSR_result1.txt')

	#print(ngram_count(data1, 2))

	autocorrelation_task(data1, period1)
	ngram_count_task(data1, len(p1), 'L1_ngrams.txt')

	print('\n==== POLYNOM 2 ====\n')

	period2 = lfsr(p2, 'LFSR_result2.txt')

	print('Period:', period2)

	data2 = import_data('LFSR_result2.txt')

	#print(ngram_count(data1, 2))

	autocorrelation_task(data2, period2)
	ngram_count_task(data2, len(p2), 'L2_ngrams.txt')

main()
```

**Висновок:** виконавши роботу, ми набули навичок у побудові лінійних регістрів зсуву з лінійним зворотним зв’язком та їх програмній реалізації; дослідили залежність лінійних рекурентних послідовностей в залежності від характеристичного полінома регістра.