

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**

**Звіт**  
**до ЛАБОРАТОРНОЇ РОБОТА №3**  
**з дисципліни: “Спеціальні розділи**  
**обчислювальної математики”**

Виконала студентка:  
групи ФІ-03  
Швець Катерина

Реалізація операцій у скінченних полях характеристики 2  
(поліноміальний базис)  
Варіант 18

## 1. Мета роботи

Одержання практичних навичок програмної реалізації обчислень у полі Галуа характеристики 2 в поліноміальному базисі; ознайомлення з прийомами ефективної реалізації критичних по часу ділянок програмного коду та методами оцінки їх ефективності.

## 2. Завдання

А) Реалізувати поле Галуа характеристики 2 степеня  $m$  в поліноміальному базисі з операціями:

- 1) знаходження константи 0 – нейтрального елемента по операції «+»;
- 2) знаходження константи 1 – нейтрального елемента по операції «\*»;
- 3) додавання елементів;
- 4) множення елементів;
- 5) обчислення сліду елемента;
- 6) піднесення елемента поля до квадрату;
- 7) піднесення елемента поля до довільного степеня (не вище  $2^m - 1$ , де  $m$  – розмірність розширення);
- 8) знаходження оберненого елемента за множенням;
- 9) конвертування (переведення) елемента поля в  $m$ -бітний рядок (строкове зображення) і навпаки, де  $m$  – розмірність розширення;

## Результати роботи:

Вхідні дані:

```
n1=PB('111100010110001011101000000110001000000101110100110100111101000010110  
011101011111110010101101011000001100011001010100101111110010100101101011000  
0001110011011010011000100001')
```

```
n2=PB('101111000111100101100001110001110010100001101110111110000011001111110  
1110000101101110110101000111000011000010010001111001010110100011011111100001  
100010000010000001111101010')
```

$$n_3=3$$
[illegible]

## Тести на коректність:

Були перевірені такі властивості:

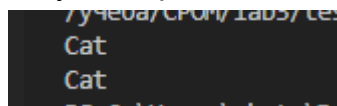
- 1)  $(a+b)*c=a*c+b*c$
- 2)  $n*n*n*...*n$  (m разів)  $= n^m$

```
from lab3 import PB
from random import choices, randint

a = PB(''.join(choices('01', k=580)))
b = PB(''.join(choices('01', k=580)))
c = PB(''.join(choices('01', k=100)))
n1=((a+b)*c).n
n2=((a*c + b*c).n)
# print(len(n1))
# print(len(a.n))
if n1 != n2:
    print("Error1")
else: print("Cat")

k = PB(''.join(choices('01', k=100)))
m=randint(1,10)
res=PB("1")
for i in range(m):
    res1=res*k
    res=res1
res2=k**m
if res1.n != res2.n:
    print("Error2")
else: print("Cat")
```

Результат роботи тестів:



```
7учеба/СРОМ/lab3/tes
Cat
Cat
PS C:\Users\cheta\de
```

Код:

```
class PB:
    m=571
    def __init__(self, n):
        p = "10000100101"
        if len(n)==self.m: self.n=n
        elif len(n)<self.m: self.n=n.zfill(self.m)
        elif len(n)>self.m:
            res = PB(n[-571:])
```

```

        for i in range(self.m, len(n)):
            if n[::-1][i] == "1":
                res += PB(p) << (i - self.m)
        try: self.n = res.n
        except: self.n = res

def __add__(self, other):
    res = ""
    for i in range(len(self.n)):
        if self.n[i] == other.n[i]: res=res+ "0"
        else: res =res+ "1"
    res=PB(res)
    return res

def __lshift__(self, n):
    res=self.n + "0" * n
    res=PB(res)
    return res

def __mul__(self, other):
    res = PB("0")
    for i in range(len(other.n)):
        if other.n[i] == "1":
            res1=self << (len(other.n) - i - 1)
            res =res1+ res
    return res

def powsq(self):
    return self*self

def __pow__(self, pow):
    res=PB("1")
    while(pow>0):
        if(pow%2==1): res=res*self
        self=self*self
        pow=pow//2
    return res

def trace(self):
    matrix=[]
    a=self.n.lstrip('0')
    for i in range(len(a)):
        matrix.append(a[i])
    return matrix[-1]

```