

Міністерство освіти і науки України

НТУУ«Київський політехнічний інститут»

Фізико-технічний інститут

Спеціальні розділи обчислювальної

Лабораторна робота

Робота №3

**Виконав:**

Студент 3 курсу ФТІ

групи ФІ-03

Буржимський Р. В

**Перевірила:**

Пекарчук Н. А.

**Реалізація операцій у скінченних полях характеристики 2**

**(поліноміальний базис)**

**МЕТА**: 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.

**Завдання. Варіант 3**

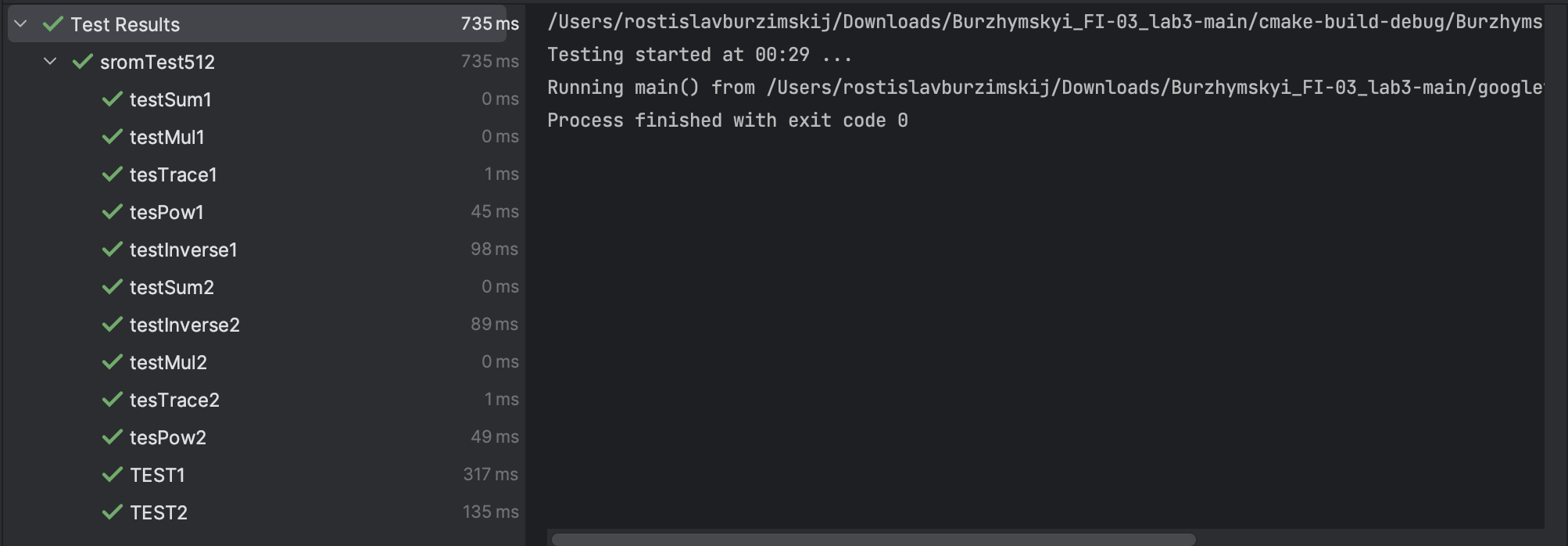
Повинні бути реалізовані такі операції:

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

Тести

#include "gtest/gtest.h"  
#include "GF2.h"  
  
  
using namespace std;  
  
TEST(sromTest512,testSum1)  
{  
 GF2 A("f01327c24a59ce5b5fef9dcb98afb2f9c18bbaadd8fa");  
 GF2 B("881b59763c21011d3c0359924251f014c4d8e9b3dcfc");  
  
 A += B;  
 string actual = A.convert64bitToHex();  
 string expected = "78087eb47678cf4663ecc459dafe42ed0553531e0406";  
  
 EXPECT\_EQ(actual,expected);  
}  
TEST(sromTest512,testMul1)  
{  
 GF2 A("f01327c24a59ce5b5fef9dcb98afb2f9c18bbaadd8fa");  
 GF2 B("881b59763c21011d3c0359924251f014c4d8e9b3dcfc");  
 GF2 res = A\*B;  
 string actual = res.convert64bitToHex();  
 string expected = "2e5779473480a9a58727a0243ceb8da95861b0d3e938";  
  
 EXPECT\_EQ(actual,expected);  
}  
TEST(sromTest512,tesTrace1)  
{  
 GF2 A("f01327c24a59ce5b5fef9dcb98afb2f9c18bbaadd8fa");  
  
 int actual = A.Trace();  
 int expected = 1;  
  
 EXPECT\_EQ(actual,expected);  
}  
TEST(sromTest512,tesPow1)  
{  
 GF2 A("f01327c24a59ce5b5fef9dcb98afb2f9c18bbaadd8fa");  
 GF2 B("881b59763c21011d3c0359924251f014c4d8e9b3dcfc");  
 GF2 res = A^B;  
 string actual = res.convert64bitToHex();  
 string expected = "194fcfb19de43f69014055f3d59fd37a87855312add95";  
  
 EXPECT\_EQ(actual,expected);  
}  
TEST(sromTest512,testInverse1)  
{  
 GF2 A("f01327c24a59ce5b5fef9dcb98afb2f9c18bbaadd8fa");  
  
 string actual = A.inverseGF().convert64bitToHex();  
 string expected = "91f1b90ffedf0421dcf799b78a89b2ac0802371030c7";  
  
 EXPECT\_EQ(actual,expected);  
}  
  
TEST(sromTest512,testSum2)  
{  
 GF2 A("5f1ab2d2f6897911a2004159cab4afcf7c4229f624cf");  
 GF2 B("9d3ca8144df69447c8b59ae9f84c8d3c7ad63f14a327");  
  
 A += B;  
 string actual = A.convert64bitToHex();  
 string expected = "c2261ac6bb7fed566ab5dbb032f822f3069416e287e8";  
  
 EXPECT\_EQ(actual,expected);  
}  
TEST(sromTest512,testInverse2)  
{  
 GF2 A("5f1ab2d2f6897911a2004159cab4afcf7c4229f624cf");  
  
 string actual = A.inverseGF().convert64bitToHex();  
 string expected = "7e77895b3d1a04f9c610a00da260a4eb75e60f163c0c3";  
  
 EXPECT\_EQ(actual,expected);  
}  
TEST(sromTest512,testMul2)  
{  
 GF2 A("5f1ab2d2f6897911a2004159cab4afcf7c4229f624cf");  
 GF2 B("9d3ca8144df69447c8b59ae9f84c8d3c7ad63f14a327");  
 GF2 res = A\*B;  
 string actual = res.convert64bitToHex();  
  
 string expected = "337d9308e8a78374ca2329bd9e769d301f390b86fae3d";  
  
 EXPECT\_EQ(actual,expected);  
}  
TEST(sromTest512,tesTrace2)  
{  
 GF2 A("5f1ab2d2f6897911a2004159cab4afcf7c4229f624cf");  
  
 int actual = A.Trace();  
 int expected = 1;  
  
 EXPECT\_EQ(actual,expected);  
}  
  
  
TEST(sromTest512,tesPow2)  
{  
 GF2 A("5f1ab2d2f6897911a2004159cab4afcf7c4229f624cf");  
 GF2 B("9d3ca8144df69447c8b59ae9f84c8d3c7ad63f14a327");  
 GF2 res = A^B;  
 string actual = res.convert64bitToHex();  
 string expected = "34aa84422df2734d6421a35270928337e2e4c50d33dc2";  
  
 EXPECT\_EQ(actual,expected);  
}  
  
TEST(sromTest512,TEST1)  
{  
 GF2 A("274f3834cd8a7bdc21b8d8748756f2b05d786631b1f2");  
 GF2 B("11f5d00d9a1eba3c3d26b75166cf6e4face044da57eb");  
 GF2 C("e22db14f661a52f154086f0dfe3594ff29420ca9542b");  
 GF2 res1(1);  
 res1+=A;  
 res1+=B;  
 res1 = res1 \* C;  
  
 GF2 res2 = A\*C;  
 res2+=(B\*C);  
 string a =res1.convert64bitToHex();  
 string e =res2.convert64bitToHex();  
 GF2 pow("7ffffffffffffffffffffffffffffffffffffffffffff");  
 A.convert64bitToHex();  
 GF2 powA = A^pow;  
 GF2 powB = B^pow;  
 GF2 powC = C^pow;  
 EXPECT\_EQ("1",powA.convert64bitToHex());  
 EXPECT\_EQ("1",powB.convert64bitToHex());  
 EXPECT\_EQ("1",powC.convert64bitToHex());  
 EXPECT\_EQ(a,e);  
}  
  
TEST(sromTest512,TEST2)  
{  
 GF2 A("cc17a88e68dc679274cb8891ba862");  
 GF2 B("acd5912bd9d706f4010f43b94c1a4");  
 GF2 C("a2e309c620d9ad66bd392554e7b25");  
 GF2 res1(1);  
 res1+=A;  
 res1+=B;  
 res1 = res1 \* C;  
  
 GF2 res2 = A\*C;  
 res2+=(B\*C);  
 string a =res1.convert64bitToHex();  
 string e =res2.convert64bitToHex();  
 GF2 pow("7ffffffffffffffffffffffffffffffffffffffffffff");  
 A.convert64bitToHex();  
 GF2 powA = A^pow;  
 GF2 powB = B^pow;  
 GF2 powC = C^pow;  
 EXPECT\_EQ("1",powA.convert64bitToHex());  
 EXPECT\_EQ("1",powB.convert64bitToHex());  
 EXPECT\_EQ("1",powC.convert64bitToHex());  
 EXPECT\_EQ(a,e);  
}

Приклад виконання тесту



TES1 TEST2 – це блокові тести на пункт Б)

**Час виконання операцій**

|  |  |  |  |
| --- | --- | --- | --- |
| Операція  Значення | Мінімальне | Середнє | Максимальне |
| Множення | 1 мс | 3 мс | 4 мс |
| Додавання | <0мс | <0мс | <0мс |
| Оберенений | 65 мс | 79 мс | 98мс |
| Піднесення до степеня | 45 мс | 56мс | 60мс |
| Трейс | 1 мс | 1 мс | 2 мс |