УДК 004.021

ПОЛТОРАК В.П.,
ВІТІЩЕНКО Н.С.

КАЛЬКУЛЯТОР GF(q) ДЛЯ ЦИКЛІЧНИХ КОДІВ

Розроблені та реалізовані алгоритми функціонування інструменту для проведення обчислень в скінченних полях Галуа. Наведені приклади його роботи. Створений інструмент виконує арифметичні операції в полях Галуа, піднесення в степінь та ділення поліномів. Він знайшов застосування в наукових дослідженнях та в учбовому процесі НТУУ «КПІ». Калькулятор може бути рекомендований в учбовий процесс інших технічних ВНЗ.

Algorithms for the functioning of the application to perform calculations in finite Galois fields are developed and implemented. Examples of its work are demonstrated. An application calculates the arithmetic GF(q) operations, does exponentiation and division of polynomials in GF(q) too. This tool is used in scientific researches and in the educational process of NTUU "KPI". The tool can be recommended for the educational process of other technical universities.

На сьогоднішній день в інформаційно-комунікаційних технологіях існує ряд задач, пов'язаних з дослідженням надлишкових завадостійких кодів, що коректують помилки. Кінцевою метою таких досліджень є вирішення задач забезпечення надійного відеоконференцзв'язку; створення системи зв'язку, як великої цифрової системи; надійної та достовірної передачі даних між обчислювальними терміналами літаючих апаратів та супутників; передача даних вузько-смуговими каналами (телефон); зберігання великого об'єму даних, чуттєвих до помилок, тощо.

Існує клас циклічних кодів, здатних вирішувати проблему з виправленням помилок «на льоту» [1]. Коди Боуза-Чоудхурі-Хоквінгема (БЧХ) здатні виправляти до t помилок в блоці даних довжини n (t<n) [2]. В якості алфавіту коду обирається множина елементів поля Галуа GF(q), де q – потужність алфавіту, завдяки чому, місце знаходження помилок можна визначити, вирішивши ряд алгебраїчних рівнянь [3].

При дослідженні шляхів вирішення таких задач, необхідно виконати багатократні процедури кодування та декодування цими кодами, вивчити вплив ряду факторів на їхню ефективність.

Відомо, що полем Галуа називають множину зі скінченним числом елементів та з заданими на ній арифметичними операціями.

Найменше поле – двійкове, воно містить два елементи, з відповідними операціями [1].

В роботі поставлена ціль – створити окремий інструмент, котрий дозволив би виконувати обчислення в скінченних полях Галуа та виконувати ділення полінома на поліном, для того, щоб підвищити швидкість моделювання та обробки недвійкових надлишкових кодів в процесі моделювання та дослідження. Крім того, велике значення має розповсюдження навичок та вмінь виконання таких робіт в кадровій підготовці кваліфікованого персоналу в IT сфері.

В теперішній час існують декілька багатофункціональних, дорогих програмних пакетів, з широким функціональним спектром, за допомогою яких можна виконати обчислення в полях Галуа. Ці програми універсальні, тому вони складні в експлуатації, оскільки вирішують велику кількість різнотипових задач. Яскравими прикладами таких пакетів є MatLab та Mathematica.

MatLab 7.0 – програмний пакет, за допомогою якого можна виконувати арифметичні операції складання «+», віднімання «-», множення «*», ділення «/» над елементами кінцевого поля (назви операцій умовні та мають свої визначення, наведені нижче) [4].

Так, наприклад, використовуючи MatLab, для рішення елементарного прикладу складання двох чисел в GF(16) необхідно виконати достатньо велику кількість натискань клавіш. При цьому, структура вводу даних досить складна.

Приклад виконання операції $(10 + 13)_{16}$ в GF(16) з використанням пакету MatLab виглядає так:

```
>> x = gf(10,4)
x = GF(2^4) array. Primitive polynomial = D^4+D+1 (19 decimal)
 Array elements = 10
```

```
>> y = gf(13,4)
 y = GF(2^4) array. Primitive polynomial = D^4+D+1 (19 decimal)
 Array elements = 13
>> z = x + y
z = GF(2^4) array. Primitive polynomial = D^4+D+1 (19 decimal)
 Array elements = 7
```

Команди, що задані після «>>» вводить користувач.

Як видно, спочатку задається перший операнд в спеціальному форматі, і його значення присвоюється змінній x. Другий операнд задається аналогічно і його значення присвоюється змінній y. Після того, як обидва операнди задані, ми створюємо третю змінну z, та їй присвоюємо результат виконання операції складання x та y.

З діленням поліномів ситуація більш складна. Спочатку необхідно задати саме скінченне поле. Воно буде відображатися в незручному для сприйняття вигляді – в не лексикографічному порядку. Елементи полінома задаються в спеціальному вигляді в командному рядочку.

Другий пакет, Wolfram Mathematica 7 - це система комп'ютерної алгебри компанії Wolfram Research. Вона також містить функції, за допомогою яких можна виконувати арифметичні операції в полях Галуа [5].

Для того, щоб виконувати ці операції спочатку необхідно підключити відповідну бібліотеку - "FiniteFields`". Далі, зіткнемось з незручністю – результати виконання операцій будуть відображатися в двійковому вигляді.

Приклад виконання операції $(10 + 13)_{16}$ в GF(16) з використанням пакету Mathematica виглядає так:

```
In[1]:=GF[2,4][{1,0,1,0}]+GF[2,4][{1,0,1,1}]
Out[1]={1,1,0,1}2.
```

Операнди задаються в спеціальному форматі. Спочатку задається поле елементів, в даному випадку $GF(2^4)$. Далі, в фігурних дужках записуємо сам операнд в двійковому вигляді. Програма повертає результат складання операндів в двійковому вигляді, що не завжди зручно для користувача.

Очевидно, що в цих програмних пакетах важко виконувати відповідну обчислювальну роботу.

Тому важливо створити зручний для оператора інструмент, в якого буде вузько направлений функціонал. Щоб за допомогою цього інструменту можна було б обробляти символи алфавіту в полях Галуа, ділити поліном на поліном в GF(q) з отриманням результатів у вигляді частки та залишку від ділення.

Для вирішення поставленої задачі необхідно розробити реалізацію арифметичних операцій складання, множення, та обернених до них операцій – віднімання, ділення, а також піднесення в степінь та ділення полінома на поліном.

В роботі розглядається підготовка поля елементів для окремого випадку БЧХ коду – коду Ріда – Соломона.

В обраній області практичне застосування мають скінченні поля, з потужністю множини, що дорівнює цілій степені двійки. Так, наприклад, елементи $GF(16) = GF(2^4)$ можуть бути подані в декількох формах (табл.1): в векторному поданні елементів поля, адитивною групою та мультиплікативною групою.

Табл. 1. Представлення поля GF(16)

N	A_N (векторне подання)	$R_m(x)$ (адитивна група)	x^i (мультиплікативна група)
0	0000	0	-
1	0001	1	x^0
2	0010	x	x^1
3	0011	x+1	x^4
4	0100	x^2	x^2
5	0101	x^2+1	x^8
6	0110	x^2+x	x^5
7	0111	x^2+x+1	x^{10}
8	1000	x^3	x^3
9	1001	x^3+1	x^{14}
10	1010	x^3+x	x^9
11	1011	x^3+x+1	x^7
12	1100	x^3+x^2	x^6
13	1101	x^3+x^2+1	x^{13}
14	1110	x^3+x^2+x	x^{11}
15	1111	x^3+x^2+x+1	x^{12}
1	0001	1	x^{15}

Розглянемо визначення та виконання вказаних вище арифметичних операцій скінченного поля, з метою підготовки їх алгоритмічної реалізації.

Складання елементів поля виконується за допомогою їх подання адитивною групою або за допомогою векторного подання (табл.1).

Наприклад,

$10 + 7 = x^3 + x + x^2 + x + 1 = x^3 + x^2 + (1 + 1)x+1= = x^3+ x^2 + 1 = 13.$

$(10 + 7)_{16} = (1010)_2 + (0111)_2 = (1101)_2 = (13)_{16}.$

Множення елементів поля виконується за допомогою їх подання мультиплікативною групою (табл.1).

Наприклад,

$10 * 7 = x^9 * x^{10} = x^{19} = x^{15} * x^4 = x^4 = 3$.

Віднімання елементів поля виконується аналогічно складанню.

Наприклад,

$10 - 7 = 10 + (-7) = 10 + 7 = x^3+x+x^2+x+1 =$

$= x^3+x^2+(1+1)x+1 = x^3+ x^2+1 = 13$.

$(10 - 7)_{16} = (10 + (-7))_{16} = (10 + 7)_{16} =$

$= (1010)_2 + (0111)_2 = (1101)_2 = (13)_{16}$.

Ділення елементів поля виконується за допомогою їх подання мультиплікативною групою, і ця операція схожа на операцію множення, тільки при діленні показники степені віднімаються (табл.1). Ділення на нуль не визначено.

Наприклад,

$10 : 7 = x^9 : x^{10} = x^{-1} = x^{15} * x^{-1} = x^{14} = 9$.

При піднесенні в степінь заданого операнду, використовується його мультиплікативне подання. При цьому степені перемножуються.

Наприклад,

$10^7 = (x^9)^7 = x^{63} = x^{15} * x^{15} * x^{15} * x^{15} * x^3 = x^3 =$

$= 8$.

Ділення полінома на поліном – більш складна процедура, яка в явному вигляді виконується за допомогою операндів, поданих поліномами. Розглянемо приклад ділення поліномів поданих у векторній формі на рис.1.

Нехай, наприклад, поліном - ділене – 1 5 8 0 0.

А поліном - дільник – 1 12 5.

В результаті виконання стандартної процедури ділення «в стовпчик» отримаємо такі результати:

Частку від ділення – 1 9 11.

Залишок – 6 1.

Проміжкові від'ємники – 1 12 5, 9 6 11, 11 13 1.

На першому кроці ділимо старший елемент діленого на старший елемент дільника та записуємо його на старшу позицію частки. Далі перемножуємо отриманий елемент частки на кожен елемент дільника, та записуємо проміжковий від'ємник під діленим. Далі, за правилами віднімання (результат якого співпадає з результатом складання для полів характеристики p = 2), додаємо перші f (f – число елементів дільника) елементів діленого з елементами проміжкового від'ємника, нижче записуємо результат.

Ці дії повторюємо до тих пір, поки в діленому ще є елементи (в даному прикладі це «0») для зносу їх до залишку. Коли порядок залишку буде меншим порядку дільника, процес ділення припиняється (тобто степінь полінома – залишку стає меншою степені полінома - дільника).

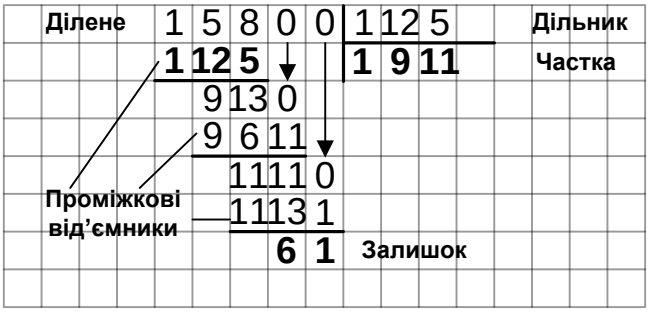

Рис. 1. Ділення поліномів в векторній формі

Інструмент повинен містити декілька обчислювальних модулів: модуль *Калькулятора*, що виконує операції складання, множення, віднімання, ділення, піднесення в степінь, а також, модуль *Дільника поліномів*, що ділить поліном на поліном.

В модулі *Калькулятора* повинен бути передбачений вибір GF(q), для $q = 2^i$. Повинні бути присутні декілька способів вводу даних - з клавіатури або за допомогою миші.

Оператор повинен мати можливість виконувати арифметичні операції з елементами скінченного поля та піднесення в степінь. Користувач повинен вводити перший операнд з клавіатури, або за допомогою миші, далі, обрати бажану операцію та вводити другий операнд. Після цього, натиснувши клавішу «=» або «Enter» на екрані повинен з'явитися покроковий результат обчислювань. При складанні та відніманні проміжковий результат повинен подаватися в двійковому вигляді. При множенні, діленні та піднесенні в степінь – в степеневому поданні елементів.

В модулі *Дільника поліномів* повинен бути передбачений ввід даних з клавіатури, а також із зовнішнього файлу (в даному випадку дані завантажуються з таблиці).

Користувач повинен задавати поліном – ділене та поліном - дільник в векторному вигляді. Після натискання клавіші «Розділити» на екрані повинен з'явитися результат ділення, у вигляді залишку та частки від ділення поліномів. В теорії завадостійкого кодування залишок від ділення поліномів цікавить нас в більшій мірі.

Оператор повинен мати можливість багатократно виконувати ділення поліномів. Для цього необхідно передбачити кнопку

«Скидання». Функціонально – логічна схема *Калькулятора* та *Дільника поліномів* зображена на рис.2.

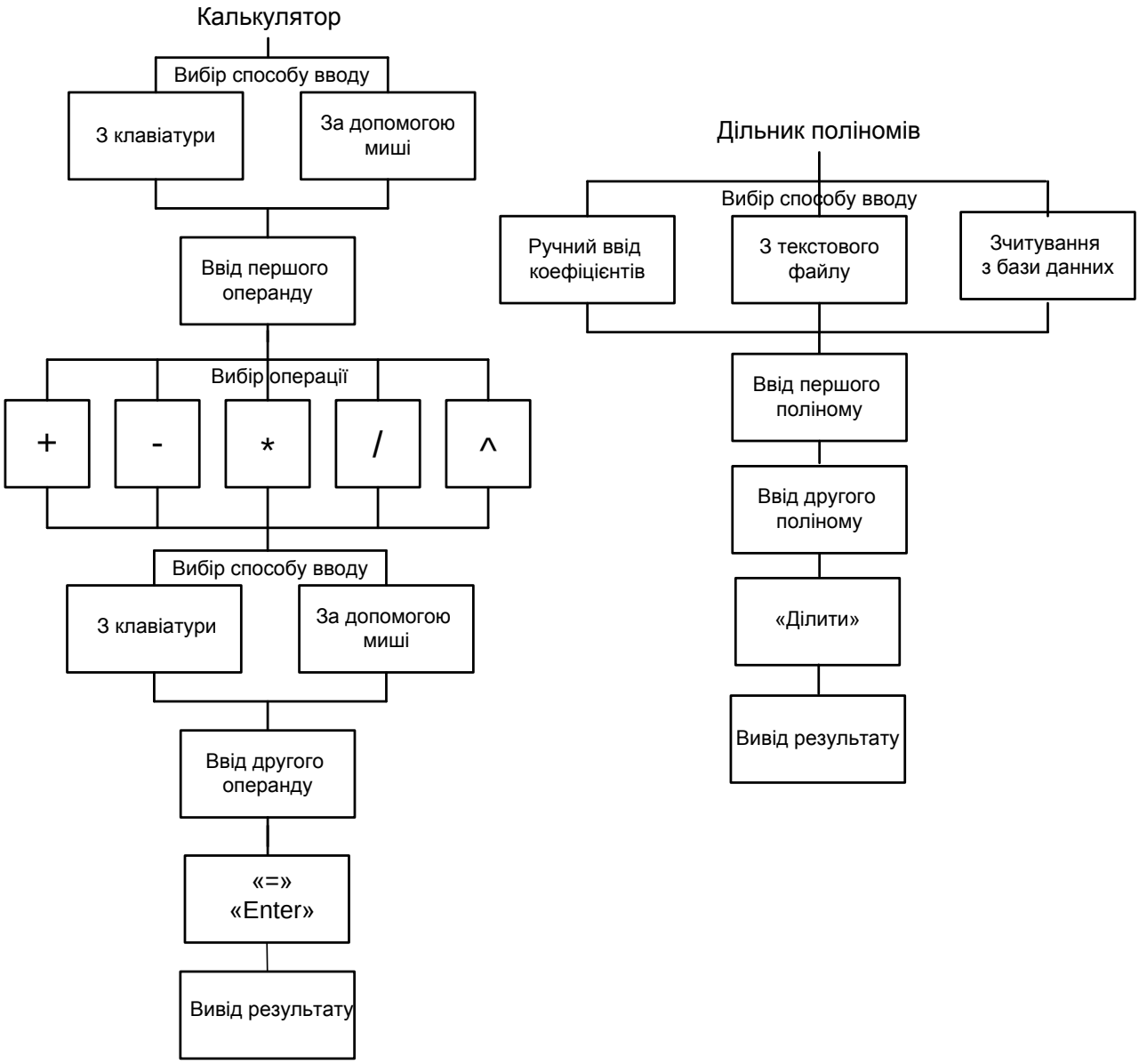

Рис. 2. Функціонально-логічна схема Калькулятора GF(q)

Для виконання операції складання двох елементів скінченного поля, необхідно використати три процедури, які описані в табл.2.

Процедура ToBin – переводить числа з q – ічного вигляду в двійковий еквівалент. Процедура DoPlus – додає два двійкових числа, за mod 2, порозрядно (без переносів). Процедура ToDec – переводить результат додавання з двійкового вигляду в q – ічний.

Табл. 2. Перелік необхідних процедур для виконання складання

ToBin	Передаємо chislo_dec – операнд в q - ічному вигляді та mas_bin – масив для зберігання двійкового числа.
ToDec	Передаємо chislo_bin – операнд в двійковому вигляді.
DoPlus	Передаємо операнд 1 та операнд 2 в двійковому вигляді; mas_result – масив для зберігання результату додавання. алг DoPlus (арг цел таб op1[1:4], цел таб op2[1:4], цел таб mas_result[1:4]) нач нц для i от 0 до 3 mas_result[i] := op1[i] ^ op2[i] кц кон

Для виконання операції множення двох елементів поля Галуа, необхідна процедура, що описана в табл.3.

Процедура DoMult – множить два операнди та вертає результат множення в q – ічному вигляді.

Табл. 3. Процедура для виконання ноження

DoMult	Передаємо операнд 1 та операнд 2; змінна result_mult – повертає результат множення.

```
алг DoMult (арг цел operand1, арг цел operand2, рез цел result_mult)
нач цел result_mult:= 0, temp := 0 | temp – змінна, в якій зберігається результат складання показників степеней двох чисел.
  если operand1 = 0 или operand2 = 0 то result := 0
  если operand1 > 0 и operand2 > 0 то
    нц temp := mas_pow[operand1] + mas_pow[operand2] | mas_pow – масив степеней для мультиплікативної групи
      если temp >= 15 то temp := temp mod 15
        нц для i от 0 до 15
          если mas_pow[i] = temp то result_mult:= i
        кц
      иначе
        нц для i от 0 до 15
          если mas_pow[i] = temp то result_mult:= i
        кц
    кц
кон
```

Операція віднімання по своїй суті аналогічна операції складання, тому в ній вико ристовуються ті ж самі процедури (табл.2).

Операція ділення елементів скінченного поля схожа на операцію множення. Різниця в тому, що показники степені ми віднімаємо (табл.4). При цьому, не існує проблеми ділення на нуль, оскільки відсутнє мультиплікативне подання для нуля.

Процедура DoDiv – ділить перший операнд на другий та повертає результат ділення.

Табл. 4. Процедура для виконання ділення

DoDiv	Передаємо операнд 1 та операнд 2; змінна result_div – повертає результат ділення.

```
алг DoDiv (арг цел operand1, арг цел operand2, рез цел result_div)
нач цел result_div:= 0 | result_div – зміна, що зберігає результат ділення двох чисел
если (operand1 != 0 и operand2 = 0) или (operand1 = 0 и operand2 = 0) то
result_div:= -1| вивід повідомлення про помилку
  если operand1 = 0 и operand2 > 0 то result_div:= 0
  если operand1 != 0 и operand2 != 0 то
    нц temp := mas_pow[operand1] - mas_pow[operand2] | temp – результат віднімання степеней операндів; mas_pow – масив степеней для мультиплікативної групи
      если temp < 0 то temp := temp + 15
        нц для i от 0 до 15 | пошук елемента, що відповідає отриманій встепені
          если mas_pow[i] = temp то result_div:= i
        кц
      иначе
        нц для i от 0 до 15
          если mas_pow[i] = temp то result_div:= i
        кц
    кц
кон
```

Піднесення в степінь здійснюється за алгоритмом, що описаний у табл.5.

Процедура DoPow – підносить перший операнд в степінь, задану другим операндом та повертає результат піднесення в степінь.

Табл. 5. Процедура піднесення в задану степінь операнда

DoPow	Передаємо операнд 1 та pow - степінь, в яку підносимо операнд 1; змінна result_pow – повертає результат піднесення в степінь.

```
алг DoPow (арг цел operand1, арг цел pow, рез цел result_pow )
нач цел result_pow:= 0 | result_pow – змінна, що зберігає результат піднесення в степінь.
  если operand1 = 0 или pow = 0 то result_pow:= 0
    иначе
      нц temp := mas_pow[operand1] * operand2; | temp – результат множення степені операнда1 на задану степінь pow.
        temp = temp mod 15
        нц для i от 0 до 15
          если mas_pow[i] = temp то result_pow:= i
        кц
      кц
кон
```

Алгоритм функціонування модуля ділення поліномів виглядає наступним чином (псевдокод):

ВХІДНІ ДАНІ:

polinom1 – строка з коефіцієнтами полінома - діленого (перший поліном).

polinom2 – строка з коефіцієнтами полінома - дільника (другий поліном).

ВИХІДНІ ДАНІ:

ostatok – масив коефіцієнтів залишку від ділення.

chast – масив коефіцієнтів частки від ділення.

Крок 1.

Перетворення коефіцієнтів першого полінома у масив елементів поля. Підрахунок кількості елементів.

Крок 2.

Перетворення коефіцієнтів другого полінома у масив елементів поля. Підрахунок кількості елементів.

Крок 3.

Записуємо частку від ділення в mas_chast[i].

Крок 4.

Множимо коефіцієнти другого полінома mas_pol_2[i] на перший елемент частки mas_chast[0], за допомогою процедури DoMult (табл.3).

Крок 5.

Додаємо елементи залишку до елементів проміжкового від'ємника mas_temp[i];

нц для і от 0 до pol_2_length | pol_2_length – кількість елементів другого полінома (полінома - дільника).

mas_ ostatok [i]:=mas_ ostatok[i]^mas_ temp [i]

кц

Крок 6.

Зсуваємо всі елементи залишку на одну позицію вліво, тим самим прибираємо нуль з першої позиції.

нц для і от 0 до mas_ostatok.length-1

mas_ ostatok [i] := mas_ ostatok [i+1]

кц

Крок 7.

Перевіряємо чи залишились ще в першому поліномі елементи, які можна знести до залишку.

Якщо залишились – зносимо елемент. Якщо ні – переходимо до кроку 8.

Крок 8.

Вивід залишку та частки від ділення у відповідні Тулбокси діалогового вікна.

При написанні *Калькулятора GF(q)* була використана мова програмування C# платформи .NET.

На рис.3 наведений вигляд головного вікна Калькулятора, наприклад, для GF (16).

За допомогою меню Калькулятора можна обрати роботу безпосередньо з модулем *Калькулятора*, або з модулем *Дільника поліномів.*

Інтерфейс *Калькулятора* містить: кнопки з цифровими значеннями, кнопки зі знаками операцій, а також область, в якій від ображається покроковий результат виконання операцій над двома операндами.

Нижче наведені приклади обчислювань Калькулятором, представлених вище в GF (16) операцій.

Тут враховано, що $x^{15} = 1$.

Так, операція (10 + 7), Калькулятором виконується, як:

$10 + 7 = 1010 + 0111 = 1101 = 13$

Операція (10 * 7) виконується, як:

$10 * 7 = x^9 * x^{10} = x^{19} = x^{15} * x^4 = 3$

Операція (10 - 7) виконується, як:

$10 - 7 = 10 + (-7) = 10 + 7 = 1010 + 0111 = 1101 = 13$

Операція (10 / 7) виконується, як:

$10 / 7 = x^9 / x^{10} = x^9 * x^{-10} = x^{-1} * x^{15} = 9$

Піднесення в степінь (10^7) виконується, як:

$10^7 = (x^9)^7 = x^{63} = x^3 = 8$

На рис.4 зображено вікно *Дільника поліномів*, наприклад, для GF (16).

Його можна запустити, обравши відповідну закладку в опціях Калькулятора GF(16). Інтерфейс вікна *Дільника поліномів* має вигляд звичний для оператора.

Поліноми у вікні дільника задаються у векторному поданні через пробіл.

В результаті ділення полінома на поліном, ми отримаємо залишок та частку від ділення.

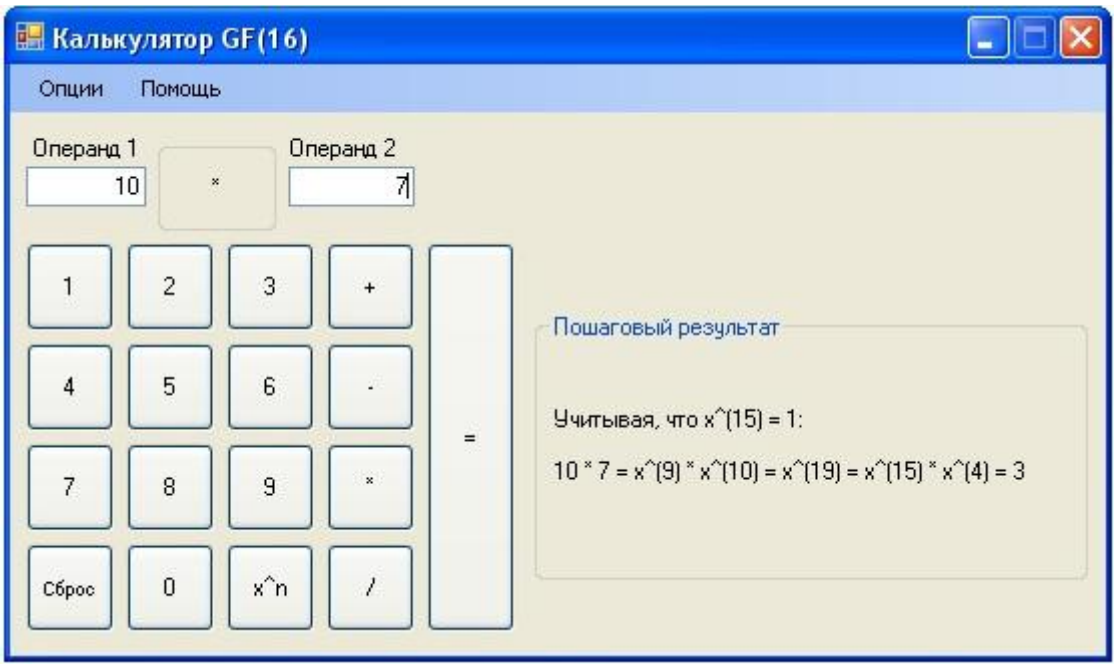

Рис. 3. Головне вікно Калькулятора GF(16)

Можна виконати необмежену кількість ітерацій ділення.

На рис.4 зображено приклад виконання ділення полінома 4 степені, поданого вектором (з набором своїх коефіцієнтів 1 5 8 0 0) на поліном 2 степені (набором своїх коефіцієнтів 1 12 5). Результат ділення відображається у вигляді залишку та частки від ділення.

		хв	**лок**
1.	Вручну	30	12
2.	З використанням *Калькулятора*	10	4
3.	З використанням *Дільника поліномів*	1	1

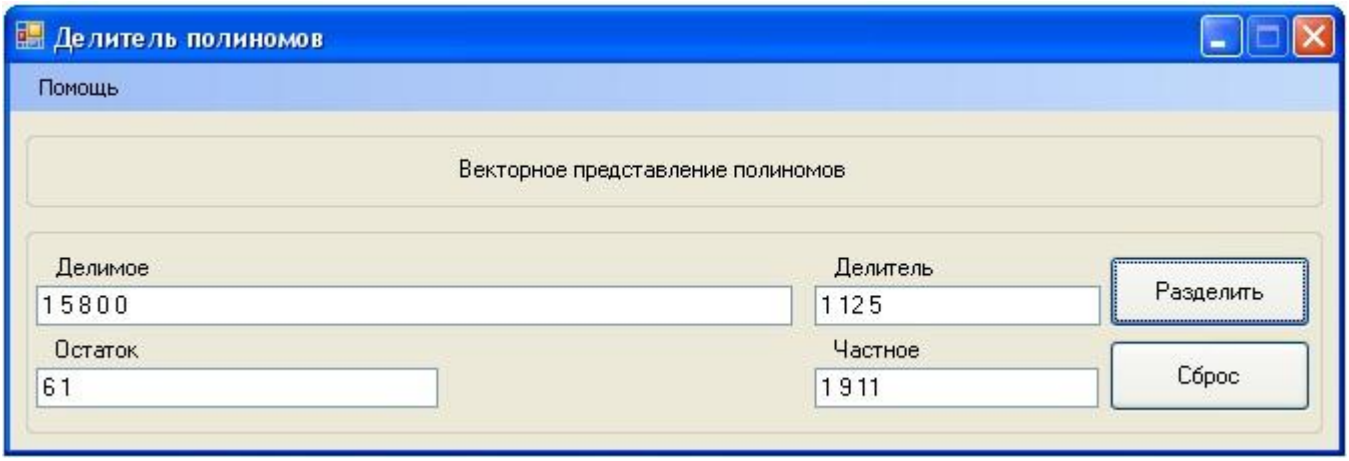

Рис. 4. Вікно Дільника поліномів

З метою перевірки ефективності створеного інструменту, були проведені натурні експерименти. Розглядалися три ситуації, в яких прийняли участь 25 студентів - операторів.

Дослід 1. Їм було доручено виконати операцію ділення полінома на поліном вручну. При цьому порядок діленого був 14, а дільника – 6. Практично всім студентам на вирішення даної задачі знадобилося близько тридцяти хвилин. Частина з них допускала помилки, в результаті чого, були отримані невірні значення розрахунків, що відображено в табл.6.

Дослід 2. На наступному етапі експерименту студенти використовували *Калькулятор* для виконання арифметичних операцій над коефіцієнтами поліномів. В цьому випадку час, який вони затратили на ділення поліномів значно зменшився. Але через людський фактор, все ще, виникали помилки в розрахунках (табл.6).

Дослід 3. На третьому етапі дослідження, обчислення проводились у *Дільнику поліномів*, на що знадобилось 1..2 хвилини щоб ввести дані, та декілька секунд, щоб отримати достовірний та правильний результат. Помилки практично були відсутні. Одна помилка виникла через неправильний ввід даних.

Усереднені результати експерименту від ображені в табл.6.

Табл. 6. Результати експерименту

Номер експерименту	**Обчислювальний тип експерименту**	**Середній час виконання,**	**Середня кількість поми-**

В рамках даної роботи під ефективністю калькулятора gf(q) будемо розуміти затрати часу т на виконання обчислювальної роботи операторами та кількість випадків помилково виконаних розрахунків.

Обчислювати ефективність е будемо, як величину обернену добутку часу т на середню кількість помилок (1) (для уникнення нескінченного значення показника ефективності, при $\varepsilon = 0$ введемо корекцію, у вигляді $\varepsilon + 1$).

$$E = \frac{1}{T \cdot (1 + \varepsilon)} \qquad (2)$$

де т – час затрачений на обчислення,

ε – Кількість виниклих помилок.

За результатами експерименту був побудований графік, що віддзеркалює ефективність застосування отриманого інструменту (рис.5).

В майбутньому планується вдосконалення калькулятора, шляхом додавання додаткового функціоналу: *перемножувач поліномів*, що перемножує поліноми, для обчислення твірного поліному g(x), циклічного коду, та застосовується у алгоритмі декодування.

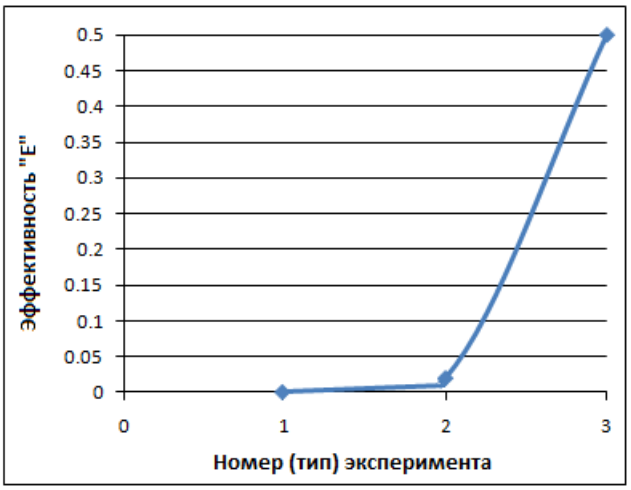

Рис. 5. Графік ефективності Калькулятора GF(q)

Обчислювач значень полінома у заданих точках.

Генератор поля, в якому задавши число елементів поля як цілу степінь двійки, можна буде генерувати подання поля у вигляді таблиці (подібній табл.1), де елементи поля будуть подані в декількох формах: десятковій, у формі залишків від ділення на твірний поліном (адитивна група), в степеневому вигляді (мультиплікативна група) і у векторній формі, а також містити відповідні елементам поля мінімальні поліноми.

Репрезентований обчислювальний інструмент використовується в наукових цілях при дослідженні властивостей надлишкових циклічних кодів.

Розроблений інструмент знайшов також застосування в учбовому процесі в НТУУ «КПІ» та може бути рекомендований для студентів інших технічних ВНЗ.

Список літератури

1. Касами Т., Токура Н., Ивадари Ё., Инагаки Я. Теория кодирования: Пер. с япон. – М.:Мир, 1978. – 568с.
2. Бэрлекэмп Э. Алгебраическая теория кодирования: Пер. с англ. – М.:Мир, 1971. – 463с.
3. Блейхут Р. Теория и практика кодов, контролирующих ошибки: Пер. с англ. – М.:Мир, 1986. – 575с.
4. Ануфриев И.Е., Смирнов А.Б., Смирнова Е.Н. MATLAB 7 (Наиболее полное руководство): БХВ-Петербург, 2005. -1104с.
5. http://www.wolfram.com/