Національний технічний університет України «Київський політехнічний інститут»

Фізико технічний інститут

Кафедра математичних методів захисту інформації МЕТОДИ РЕАЛІЗАЦІЇ КРИПТОГРАФІЧНИХ МЕХАНІЗМІВ

Лабораторна робота №1

Тема: “Вибір та реалізація базових фреймворків та бібліотек”

Виконали:

Морозюк Анастасія,

Гетьман Дмитро,

Мітрофанова Еліна

Перевірила:

Селюх П.В.

Київ – 2021

Мета роботи:

Вибір базових бібліотек/сервісів для подальшої реалізації криптосистеми.

Завдання: дослідження алгоритмів реалізації арифметичних операцій над великими (багато розрядними) числами над скінченими полями та групами з точки зору їх ефективності за часом та пам'яттю для різних програмно-апаратних середовищ.

Бібліотеки багаторозрядної арифметики, вбудовані в програмні платформи C++/C# (BigInteger), Java (BigInt) та Python (обрати одну з них) для процесорів із 32-розрядною архітектурою та обсягом оперативної пам'яті до 8 ГБ (робочі станції)

Конструктори:

BigInteger(Byte[])	Ініціалізує новий екземпляр структури BigInteger, використовуючи значення у масиві типу byte.
BigInteger(Decimal)	Ініціалізує новий екземпляр структури BigInteger, використовуючи значення Decimal.
BigInteger(Double)	Ініціалізує новий екземпляр структури BigInteger, використовуючи значення з плаваючою комою подвійної точності.
BigInteger(Int32)	Ініціалізує новий екземпляр структури BigInteger, використовуючи 32-розрядне ціле число.
BigInteger(Int64)	Ініціалізує новий екземпляр структури BigInteger, використовуючи 64-розрядне ціле число зі знаком.
BigInteger(Single)	Ініціалізує новий екземпляр структури BigInteger, використовуючи значення з плаваючою комою одиночної точності.
BigInteger(UInt32)	Ініціалізує новий екземпляр структури BigInteger за допомогою 32-розрядного цілого числа без знака.
BigInteger(UInt64)	Ініціалізує новий екземпляр структури BigInteger, використовуючи 64-розрядне ціле без знака.
BigInteger(ReadOnlySpan<Byte>, Boolean, Boolean)	Ініціалізує новий екземпляр структури BigInteger, використовуючи значення в

	діапазоні байтів, доступному тільки для читання, і при необхідності вказуючи кодування підпису та порядок байтів.

Властивості:

IsEven	Вказує, чи значення поточного об'єкта BigInteger дорівнює парному числу.
IsOne	Вказує, чи значення поточного об'єкта BigInteger має значення One.
IsPowerOfTwo	Вказує, чи значення поточного об'єкта BigInteger ступеня двох.
IsZero	Вказує, чи значення поточного об'єкта BigInteger значення Zero.
MinusOne	Отримує значення, яке становить мінус одиницю (-1).
One	Отримує значення, що становить одиницю (1)..
Sign	Отримує число, що вказує на знак (мінус, плюс або нуль) поточного об'єкта BigInteger.
Zero	Отримує значення, що становить 0 (нуль).

Методи:

Abs(BigInteger)	Отримує повне значення об'єкта BigInteger.
Add(BigInteger, BigInteger)	Складає два значення BigInteger та повертає результат.
Compare(BigInteger, BigInteger)	Порівнює два значення BigInteger і повертає ціле значення, яке показує, більше або менше перше значення порівняно з другим або дорівнює йому.
CompareTo(BigInteger)	Порівнює даний екземпляр з іншим екземпляром BigInteger і повертає ціле число, яке показує, чи значення даного

	екземпляра є меншим, більшим або рівним значенню зазначеного об'єкта.
CompareTo(Int64)	Порівнює даний екземпляр з 64-розрядним знаковим цілим числом і повертає ціле число, яке показує, чи значення даного екземпляра є меншим, більшим або рівним значенню 64-бітового знакового цілого числа.
CompareTo(Object)	Порівнює даний екземпляр із зазначеним об'єктом і повертає ціле число, яке показує, чи є значення даного екземпляра менше, більше або дорівнює значенню заданого об'єкта.
CompareTo(UInt64)	Порівнює даний екземпляр з 64-розрядним цілим числом без знака і повертає ціле число, яке показує, чи значення даного екземпляра є меншим або більшим у порівнянні зі значенням 64-бітового цілого числа без знака або рівним йому.
Divide(BigInteger, BigInteger)	Поділяє одне значення BigInteger на інше і повертає результат.
DivRem(BigInteger, BigInteger, BigInteger)	Поділяє одне значення BigInteger на інше, повертає результат, а також повертає залишок у вигляді параметра виводу.
Equals(BigInteger)	Повертає значення, що визначає чи поточний екземпляр і вказаний об'єкт BigInteger.
Equals(Int64)	Повертає значення, що визначає, чи рівні поточний екземпляр та 64-розрядне ціле число зі знаком.
Equals(Object)	Повертає значення, яке визначає, чи рівні поточний екземпляр та вказаний об'єкт.
Equals(UInt64)	Повертає значення, що визначає, чи рівні поточний екземпляр та 64-розрядне ціле число без знака.

GetBitLength()	Повертає число біт, необхідних для найкоротшого представлення поточного екземпляра додатковому коді без знакового біта.
GetHashCode()	Повертає хеш-код для поточного об'єкта BigInteger.
GetByteCount(Boolean)	Повертає число байтів, яке буде виводитися ToByteArray(Boolean, Boolean) та TryWriteBytes(Span<Byte>, Int32, Boolean, Boolean).
GreatestCommonDivisor(BigInteger, BigInteger)	Знаходить найбільший спільний дільник двох значень BigInteger.
Log(BigInteger)	Повертає натуральний логарифм (з основою e) вказаного числа.
Log(BigInteger, Double)	Повертає логарифм зазначеного числа у системі числення із зазначеною основою.
Max(BigInteger, BigInteger)	Повертає найбільше із двох значень BigInteger.
Min(BigInteger, BigInteger)	Повертає найменше із двох значень BigInteger.
ModPow (BigInteger, BigInteger, BigInteger)	Виконує модульний розподіл числа, зведеного в міру іншого числа.
Multiply(BigInteger, BigInteger)	Повертає добуток двох значень BigInteger.
Negate(BigInteger)	Змінює знак вказаного BigInteger.
Parse(ReadOnlySpan<Char>, NumberStyles, IFormatProvider)	Перетворює представлення числа, що міститься у вказаному діапазоні символів тільки для читання у вказаному стилі на його еквівалент BigInteger.
Pow(BigInteger, Int32)	Зводить значення BigInteger у заданий рівень.
Remainder (BigInteger, BigInteger)	Виконує цілий поділ двох значень BigInteger і повертає залишок.

Subtract(BigInteger, BigInteger) ToByteArray()	Віднімає одне значення BigInteger з іншого та повертає результат. Перетворює значення BigInteger на масив байтів
ToString()	Перетворює числове значення поточного об'єкта BigInteger на еквівалентне йому рядкове уявлення.
TryFormat(Span<Char>, Int32, ReadOnlySpan<Char>, IFormatProvider)	Форматує цей екземпляр довгого цілого числа діапазон символів.
TryParse(ReadOnlySpan<Char>, BigInteger)	Намагається перетворити уявлення числа, що міститься у вказаному діапазоні символів тільки для читання, в його еквівалент типу BigInteger і повертає значення, що визначає, успішно виконано перетворення.

.

Більше методі можна знайти в офіційній докуметації.

Оператори:

Addition(BigInteger, BigInteger)	Складає значення двох зазначених об'єктів BigInteger.
BitwiseAnd(BigInteger, BigInteger)	Виконує бітову операцію And для двох значень BigInteger.
BitwiseOr(BigInteger, BigInteger)	Виконує бітову операцію Or для двох значень BigInteger.
Decrement(BigInteger)	Зменшує значення BigInteger на 1.
Division(BigInteger, BigInteger)	Поділяє вказане значення BigInteger на інше вказане значення BigInteger, використовуючи цілісний поділ.
Equality(BigInteger, BigInteger)	Повертає значення, яке вказує, чи рівні значення двох об'єктів BigInteger.

Equality(BigInteger, Int64)	Повертає значення, яке вказує, чи рівні значення BigInteger і довге ціле число знакового
ExclusiveOr(BigInteger, BigInteger)	Виконує бітову операцію виключає Or (XOr) для двох значень BigInteger.
Explicit(BigInteger to Byte)	Визначає явне перетворення об'єкта BigInteger на байтове значення без знака.
GreaterThan(BigInteger, BigInteger)	Повертає значення, що визначає, чи значення типу BigInteger дійсно більше іншого значення типу BigInteger.
Implicit(Byte to BigInteger)	Визначає неявне перетворення значення типу byte без знака значення BigInteger.
Increment(BigInteger)	Збільшує значення BigInteger на 1.
. Inequality(BigInteger, BigInteger)	Повертає значення, яке вказує, чи мають два об'єкти BigInteger різні значення.
LeftShift(BigInteger, Int32)	Зсуває значення BigInteger на вказану кількість бітів вліво
LessThan(BigInteger, BigInteger)	Повертає значення, що визначає, чи значення типу BigInteger дійсно менше іншого значення типу BigInteger.
LessThanOrEqual(BigInteger, BigInteger)	Повертає значення, що визначає, чи значення типу BigInteger дійсно менше або дорівнює іншому значенню типу BigInteger.
Modulus (BigInteger, BigInteger)	Повертає залишок від поділу двох заданих значень BigInteger.
Multiply(BigInteger, BigInteger)	Помножує два задані значення BigInteger.
OnesComplement(BigInteger)	Повертає результат бітової операції доповнення до одиниці значення BigInteger.
RightShift(BigInteger, Int32)	Зсуває значення BigInteger на вказану кількість бітів праворуч.

Subtraction(BigInteger, BigInteger)	Віднімає значення BigInteger з іншого значення BigInteger.
UnaryNegation(BigInteger)	Змінює знак вказаного BigInteger.
UnaryPlus(BigInteger)	Повертає значення операнда BigInteger. (Знак операнда не змінюється.)

Більше операторів можна знайти в офікій документації.

Хід роботи

В ході роботи ми розглядали різні методи, конструктори та оператори з класу BigInteger . Ми генерували великі числа шляхом генерування випадкового масиву байтів та за допомогою конструктора BigInteger(Byte[]) створювався BigInteger. З утвореними великими числами було проведено різні операції та її порівняння.

Великі числа C# можна створювати за допомогою методу Parse - перетворює рядкове представлення числа його еквівалент типу BigInteger. Що ми також зробили, але в цьому випадку можуть визникати додаткові труднощі.

Дуже часто великі числа для подальшої роботи з ними находять у форматі Hex саме тому ми написали метод ParseHex, який приймає на вхід стрінг у форматі Hex, та повертає у десятковому аби продовжувати роботу з ним.
Оскільки при роботі зі стандартним методом Parse(String, IFormatProvider) ми зістовшнулися з певними труднощями, а саме дуже великі числа поверталися негативними використовуючи цей метод. Ось приклад роботи: Було задано однакові числа в вигляді Hex та з них створенено нові екземпляри класу BigInteger, але результат вийшов різний:

```
input: A12497B017796C126D89EAE27D6A7A752123FFF2167A6767DDCBE67EE67A12497B017796C126D89EAE27D6A7A752123FFF2167A6767DDCBE67EE67
number1: 767589932259778568435588194813714866732818887617397808001364757565982013010239306238070638333401433775453772908256809144368599213339142939811 9
number2: -451843095207405896947848223074240721450227593866409759646439447393537939152614688924843828319823205221808083358859657159223266936767025984757 7
```

Така поведінка може значно вплинути на результат роботи, наприклад якщо ми захочемо піднести одне велике число у степінь іншого великого числа, то ми отрумаємо виключення якщо показних степеню буде від’ємним.

Також ми самостійно написали функцію герування масиву байтів заданої довжини GenerateRandomByteSeed та ствоюємо за допомогою масивів нові великі числа BigInteger для подальшої роботи.

Таким чином для різної байтової довжини, в нашому випадку 64, 128 та 256 було створено пару чисел та виконано різні арифметичні операції, а саме додавання, віднімання, множення та діленя за допомогою відповідних операторів та методів класу BigInteger. Нижче наведено скріншоти з прикладом виконання, як і очукувалось результат виконання арифметичної операції за допомогою методу та оператора однаковий, проте час виконання різний. Нижче наведено таблицу часу роботи операції для різної довжини.

	64 байта	128 байтів	256 байтів
Добавання метод	00:00:00.0046867	00:00:00.0000100	00:00:00.0000510
Додавання оператор	00:00:00.0000052	00:00:00.0000502	00:00:00.0000173
Віднімання метод	00:00:00.0000316	00:00:00.0000106	00:00:00.0000104
Віднімання оператор	00:00:00.0000173	00:00:00.0000292	00:00:00.0000120
Множення метод	00:00:00.0001584	00:00:00.0007510	00:00:00.0000416
Множення оператор	00:00:00.0000133	00:00:00.0001731	00:00:00.0000261
Ділення метод	00:00:00.0001075	00:00:00.0000143	00:00:00.0000116
Ділення оператор	00:00:00.0000130	00:00:00.0000122	00:00:00.0002398
Піднесення до степеня метод	00:00:00.0002630	00:00:00.0000305	00:00:00.0000707

Піднесення до степеня оператор	00:00:00.0000410	00:00:00.0000168	00:00:00.0000397

Результати роботи програми:

```
input: A12497B017796C126D89EAE27D6A7A752123FFF2167A6767DDCBE67EE67A12497B017796C126D89EAE27D6A7A752123FFF2167A6767DDCBE67EE67
number1: 76758993225977856843558819481371486673281888761739780800136475756598201301023930623807063833340143377545377290825680914436859921333914293981 19
number2: -45184309520740589694784822307424072145022759386640975964643944739353793915261468892484382831982320522180808335885965715922326693676702598475 77
================================================================================================================================================================
   Numbers creation:
Byte length: 64
number1: 0D104CEA2474F4115BEFD21AB1F502060A187D1CEF4CEF74187DAE9F562385D9CAA58E6EAD858663FD77572952B8AC68A7655ABBF7A885EB7EEA7FAAD8E198A
number2: 0B8563529CAEE17CAF88ECB9F063612959536A3D76FD40C9D483A7B22EA7AEE8DF560AED715A68F811066628CDE3EF333E91F275F35348E662D4BD05A702731

   ------------------------------------------------------------------------

   Sum of 2 numbers using method vs operator:
Method:
Sum: 1895B03CC123D58E0B78BED4A258632F636BE75A664A303DED01565184CB34C2A9FB995C1EDFEF5C0E7DBD52209C9B9BE5F74D31EAFBCED1E1BF3CB07FE40BB
Time: 00:00:00.0046867
Operator:
Sum: 1895B03CC123D58E0B78BED4A258632F636BE75A664A303DED01565184CB34C2A9FB995C1EDFEF5C0E7DBD52209C9B9BE5F74D31EAFBCED1E1BF3CB07FE40BB
Time: 00:00:00.0000052

   ------------------------------------------------------------------------

   Subtraction of 2 numbers using method vs operator:
Method:
Subtraction: 1895B03CC123D58E0B78BED4A258632F636BE75A664A303DED01565184CB34C2A9FB995C1EDFEF5C0E7DBD52209C9B9BE5F74D31EAFBCED1E1BF3CB07FE40BB
Time: 00:00:00.0000316
Operator:
Subtraction: 1895B03CC123D58E0B78BED4A258632F636BE75A664A303DED01565184CB34C2A9FB995C1EDFEF5C0E7DBD52209C9B9BE5F74D31EAFBCED1E1BF3CB07FE40BB
Time: 00:00:00.0000173

   ------------------------------------------------------------------------

   Division of 2 numbers using method vs operator:
Method:
Division: 1
Time: 00:00:00.0001075
Operator:
Division: 1
Time: 00:00:00.0000130
```

```
   Multiplication of 2 numbers using method vs operator:
Method:
Multiplication: 09681D78A2C01CD19A12EBAD8D4D026ACDEA22AF796AE8003BA4E319B8EF35DFCFA1980D5403D487B9FF0609109C8974F6BD4A25C64CD05
E1CF353F718843AC2740D806C86685CE341A6BAD547795B7CE5DC7DF967A639416300CADA2BC718C5CD51F5286F369D135C4D45C72CCC443BFBE8AAE3F26D74
D2768B18E76E96A
Time: 00:00:00.0001584
Operator:
Multiplication: 09681D78A2C01CD19A12EBAD8D4D026ACDEA22AF796AE8003BA4E319B8EF35DFCFA1980D5403D487B9FF0609109C8974F6BD4A25C64CD05
E1CF353F718843AC2740D806C86685CE341A6BAD547795B7CE5DC7DF967A639416300CADA2BC718C5CD51F5286F369D135C4D45C72CCC443BFBE8AAE3F26D74
D2768B18E76E96A
Time: 00:00:00.0000133

   ------------------------------------------------------------------

   Power of number using method vs operator:
Method:
Power: 08B570DF1BFE6D9999B4A959B6681AC38E2515A22D63FE83CE60A4E3F1B333D199858A36E5FB05FC2FECA291C7FB7D7028FB03117BC7838A52736317
D4F5CBA40B0882D21941E8531FCD9DC5F39CEFAE35CD2A8147AB3217D5D391271AA795AE49BF4C07C153288465237A3D41F0D58162B04AFAD5170F0B3F6CBCF
4D18D8B0E11B9613BE3364D706E1858ADAEBC080234FEB407BF8BC188D6BEB5EDE5832C7AF9485998997F3BCCDF672BEB652E48FB58C4AC935624DFB820DA01
965E8
Time: 00:00:00.0002630
Operator:
Power: 08B570DF1BFE6D9999B4A959B6681AC38E2515A22D63FE83CE60A4E3F1B333D199858A36E5FB05FC2FECA291C7FB7D7028FB03117BC7838A52736317
D4F5CBA40B0882D21941E8531FCD9DC5F39CEFAE35CD2A8147AB3217D5D391271AA795AE49BF4C07C153288465237A3D41F0D58162B04AFAD5170F0B3F6CBCF
4D18D8B0E11B9613BE3364D706E1858ADAEBC080234FEB407BF8BC188D6BEB5EDE5832C7AF9485998997F3BCCDF672BEB652E48FB58C4AC935624DFB820DA01
965E8
Time: 00:00:00.0000410
================================================================================================================================
====================

   Numbers creation:
Byte length: 128
number1: 0EB0AA1C8638AC058A20C4E436805B4C8C0E0117380ADF5D683F15342583B4C20725CE23427814925740B2BF8013D00C16E070EB66EF570CE3FC84
AADB03564CD05508068C722D0CE977F69F07C359ECE9E0378BC095270F2C5AA57F74FE1A5DEFCA72D9F0B9EC5E6C41708DEE8357997BF9A8E5513607532EF8C
078D0994E5
number2: 0C1457C4ECBFA633FCCFE36DBE43C23A9323E0A1B1731A0E399BF136CF9A044AE013B89E9F8F44406A1C43D95E57FD7B10994FDE07DADCE4A2E754
20F01050409941B5ABBEE2C3D2F1E87C9E67CCF58DCCF1472FDAAD9CA5CF677163624638D6BAAB9653D5BA6C67B91A7BA4709C5C145AC66738F70AFC3F959B6
61B4A8CFD7

   ------------------------------------------------------------------
```

```
   Sum of 2 numbers using method vs operator:
Method:
Sum: 1AC501E172F8523986F0A851F4C41D871F31E1B8E97DF96BA1DB066AF51DB90CE73986C1E20758D2C15CF698DE6BCD872779C0C96ECA33F186E3D8CBCB
13A68D6996BDB24B54F0DFDB60733D6F904F7AB6D17EBB9B42C3B4FBC216E2D7445334AA76092DC67458C6255BEC325F1FB3ADD6C0101E48410392C49426941
B264BC
Time: 00:00:00.0000100
Operator:
Sum: 1AC501E172F8523986F0A851F4C41D871F31E1B8E97DF96BA1DB066AF51DB90CE73986C1E20758D2C15CF698DE6BCD872779C0C96ECA33F186E3D8CBCB
13A68D6996BDB24B54F0DFDB60733D6F904F7AB6D17EBB9B42C3B4FBC216E2D7445334AA76092DC67458C6255BEC325F1FB3ADD6C0101E48410392C49426941
B264BC
Time: 00:00:00.0000502

   ------------------------------------------------------------------

   Subtraction of 2 numbers using method vs operator:
Method:
Subtraction: 1AC501E172F8523986F0A851F4C41D871F31E1B8E97DF96BA1DB066AF51DB90CE73986C1E20758D2C15CF698DE6BCD872779C0C96ECA33F186
E3D8CBCB13A68D6996BDB24B54F0DFDB60733D6F904F7AB6D17EBB9B42C3B4FBC216E2D7445334AA76092DC67458C6255BEC325F1FB3ADD6C0101E48410392C
49426941B264BC
Time: 00:00:00.0000106
Operator:
Subtraction: 1AC501E172F8523986F0A851F4C41D871F31E1B8E97DF96BA1DB066AF51DB90CE73986C1E20758D2C15CF698DE6BCD872779C0C96ECA33F186
E3D8CBCB13A68D6996BDB24B54F0DFDB60733D6F904F7AB6D17EBB9B42C3B4FBC216E2D7445334AA76092DC67458C6255BEC325F1FB3ADD6C0101E48410392C
49426941B264BC
Time: 00:00:00.0000292

   ------------------------------------------------------------------

   Division of 2 numbers using method vs operator:
Method:
Division: 1
Time: 00:00:00.0000143
Operator:
Division: 1
Time: 00:00:00.0000122
```

```
   Multiplication of 2 numbers using method vs operator:
Method:
Multiplication: 0B172CFF72CF01C79C884E87CDBC0240022A9A4A486AFD6762C868CF4AD86E2D5D718A74C4AFED52C6CA913DDE89862D985121F6CD99E6C
63245C7BEBF08CD24F06AF7121D22689470E5D9AD659BF16969A362604E8144668169874337B527EF355CD91B642E9EA120AC94E5819DA1A0800B4EA5A4B988
B3D8C02BADEBF312A1E03C55DA343A88A2B83562C85D1200BF4E9E5A7654426F5953C0C752232F616343D81CA16D68F3FC7663FE2ACA514BA74F6998638B69E
BEE4675FC1BE045C2A43D9905063F29A9F1CFAAF9C26A37B0D7A4E06F67F2DA7369A2C0D73ABA944660E4DF34F54CF02FE4659610E3B201E6315EE5D7E635EF
E6EA13921ECB93753
Time: 00:00:00.0007510
Operator:
Multiplication: 0B172CFF72CF01C79C884E87CDBC0240022A9A4A486AFD6762C868CF4AD86E2D5D718A74C4AFED52C6CA913DDE89862D985121F6CD99E6C
63245C7BEBF08CD24F06AF7121D22689470E5D9AD659BF16969A362604E8144668169874337B527EF355CD91B642E9EA120AC94E5819DA1A0800B4EA5A4B988
B3D8C02BADEBF312A1E03C55DA343A88A2B83562C85D1200BF4E9E5A7654426F5953C0C752232F616343D81CA16D68F3FC7663FE2ACA514BA74F6998638B69E
BEE4675FC1BE045C2A43D9905063F29A9F1CFAAF9C26A37B0D7A4E06F67F2DA7369A2C0D73ABA944660E4DF34F54CF02FE4659610E3B201E6315EE5D7E635EF
E6EA13921ECB93753
Time: 00:00:00.0001731

   ----------------------------------------------------------------------

   Power of number using method vs operator:
Method:
Power: 0C621B4E1F12F1306820F3A40DF6013D701D5777E3AEA1C9EA0828E2745FE33D70C9A27A25E8CEBD519AB825DD98B80F357CDF49C67F48BA6261ABC8
B33CA9737F8717EB8DB7BD22F1FCB3E596280F2C52DF3D32D89BBA8B1EF2671321C72AD7BECDACB7265C2F374C1606BB95BFA2A8329D46CCBB2A342A7241D63
A28833A4777F0CE46E874B635CCA27BDFC9EC5E7985AC270F6C154506A9FE06607FC44DCED03E03305563DB03884DC526BF0C0B69C101AC7A7D7E140BB53F2D
B818AEE7A45C74F0241A03E73C604DCE4608AC1F1C917DDB5AF1BDDDFE6C5C6B3DE7BAB6B352A58480D373829DC8BD8F702BA1AE146F9879AF0861A0273B6B6
CACB9BA9ED8FB68968311C63F8338C1D5B56639BBB88784E192660EE54BDC40629E0EC5740AA62696AB8F6D0F2CF77640EAAE24061601E172E2BA52A4D0CF09
D80C77521B1A7F4A54485A709071517D030BB229CFE18D137CA5CDC4BE8FC7173BAC413F697E43CC0473E3607B906D3E3E9ADA5CE904827941FA46943359EAF
83529A1D
Time: 00:00:00.0000305
Operator:
Power: 0C621B4E1F12F1306820F3A40DF6013D701D5777E3AEA1C9EA0828E2745FE33D70C9A27A25E8CEBD519AB825DD98B80F357CDF49C67F48BA6261ABC8
B33CA9737F8717EB8DB7BD22F1FCB3E596280F2C52DF3D32D89BBA8B1EF2671321C72AD7BECDACB7265C2F374C1606BB95BFA2A8329D46CCBB2A342A7241D63
A28833A4777F0CE46E874B635CCA27BDFC9EC5E7985AC270F6C154506A9FE06607FC44DCED03E03305563DB03884DC526BF0C0B69C101AC7A7D7E140BB53F2D
B818AEE7A45C74F0241A03E73C604DCE4608AC1F1C917DDB5AF1BDDDFE6C5C6B3DE7BAB6B352A58480D373829DC8BD8F702BA1AE146F9879AF0861A0273B6B6
CACB9BA9ED8FB68968311C63F8338C1D5B56639BBB88784E192660EE54BDC40629E0EC5740AA62696AB8F6D0F2CF77640EAAE24061601E172E2BA52A4D0CF09
D80C77521B1A7F4A54485A709071517D030BB229CFE18D137CA5CDC4BE8FC7173BAC413F697E43CC0473E3607B906D3E3E9ADA5CE904827941FA46943359EAF
83529A1D
Time: 00:00:00.0000168
================================================================================================================================
```

```
   Numbers creation:
Byte length: 256
number1: 5B7FCAFD267AC164E97F28FD4651A009A078B9182F8746A2B1D46051E87E14A3F3C8BA1EC24B611A47F95EBA9806E19E0D3534FEB3754E169E8D49
8895085D4739C00428F0AE3ED4766C8D7D9EE42FFA5CF65946C64E64F7F1ABCC0020D887D577FC9EC7013E94677BFA4C176C3FC16A8D69601CDDD544D888056
68415B9CFC383FABD5F4FBD64BD12371B0A26ECFD5730E2BAF09B0ACF345CED8565FBA79F960573046A97F14D4859FC27AD09A0C50D6B622CB450D537876454
519C05BDB93AE424AAECFA68B95C970AC5D2923388AD898A22B4E184BBB060FED2514EE4531ECFBAE0924E4F988D160E380C85978637A5A08058DC88F4C5B7F
93B92F6D06A
number2: 1789CB25944D67F27A39AB3FA2AEF51CFC1C5C67EF5890BF320D053B2956F5EFD65A4BC6A4F4C58B3EE15044E56AD5E1980D905F955580624F701A
5AD9A5D4CA7D90AA77908E2A8D7CF1BD6FBBE0304D67DEE1FE21738080D1DB54CF81417DFB510B2415B73DC07A0BB2D215D6513EA43F9C9C56A125D6A2A661B
45009363F4E0793C1F9846FE904358223F71526E2862B5E05C190848E1204A985A141F5FD2FAAE83CB8D86902445D2DDA3037DB4C2B02F6450665E937434C9A
6287B4463110B25BD36152E47D70646BAAC8C7DBAC673039E77305AF8B7D498E9D56DA66AF08F5A20BE4437C64EA694AB6F16681D1E9F03DEE634F26B2BE401
6DEB6CFE7BB

   ----------------------------------------------------------------------

   Sum of 2 numbers using method vs operator:
Method:
Sum: 73099622BAC8295763B8D43CE90095269C9515801EDFD761E3E1658D11D50A93CA2305E5674026A586DAAEFF7D71B77FA542C55E48CACE78EDFD63E36E
AE3211B750AEA0813C6961F35E4AED5AC46047C4D53B44E7C1E578C38720CFA21A05D0C907C2DCB87C54E187AD1E2D4291000ECD05FC737EFB1B7B2E671AD41
EF00F118B8E7F58D42D4DC147B93F013C13DFDD5C40C0B22B8F5D4661970B073D9D9CC5B05B4123705A4F8CB72A01DD417C11386E5871BAB6BE6ECAB0EEB423
BA03EA4B96807E4E4D4D36CCFB76709B5A0F3514B9C40A27E734472DAA8D6FA8294B0227C55CEC7691CBFD777F58EEFDEC19582195DE6EBC2BAFA783F8101A4
9C6B825
Time: 00:00:00.0000510
Operator:
Sum: 73099622BAC8295763B8D43CE90095269C9515801EDFD761E3E1658D11D50A93CA2305E5674026A586DAAEFF7D71B77FA542C55E48CACE78EDFD63E36E
AE3211B750AEA0813C6961F35E4AED5AC46047C4D53B44E7C1E578C38720CFA21A05D0C907C2DCB87C54E187AD1E2D4291000ECD05FC737EFB1B7B2E671AD41
EF00F118B8E7F58D42D4DC147B93F013C13DFDD5C40C0B22B8F5D4661970B073D9D9CC5B05B4123705A4F8CB72A01DD417C11386E5871BAB6BE6ECAB0EEB423
BA03EA4B96807E4E4D4D36CCFB76709B5A0F3514B9C40A27E734472DAA8D6FA8294B0227C55CEC7691CBFD777F58EEFDEC19582195DE6EBC2BAFA783F8101A4
9C6B825
Time: 00:00:00.0000173

   ----------------------------------------------------------------------

   Subtraction of 2 numbers using method vs operator:
Method:
Subtraction: 73099622BAC8295763B8D43CE90095269C9515801EDFD761E3E1658D11D50A93CA2305E5674026A586DAAEFF7D71B77FA542C55E48CACE78ED
FD63E36EAE3211B750AEA0813C6961F35E4AED5AC46047C4D53B44E7C1E578C38720CFA21A05D0C907C2DCB87C54E187AD1E2D4291000ECD05FC737EFB1B7B2
E671AD41EF00F118B8E7F58D42D4DC147B93F013C13DFDD5C40C0B22B8F5D4661970B073D9D9CC5B05B4123705A4F8CB72A01DD417C11386E5871BAB6BE6ECA
B0EEB423BA03EA4B96807E4E4D4D36CCFB76709B5A0F3514B9C40A27E734472DAA8D6FA8294B0227C55CEC7691CBFD777F58EEFDEC19582195DE6EBC2BAFA78
3F8101A49C6B825
Time: 00:00:00.0000104
Operator:
Subtraction: 73099622BAC8295763B8D43CE90095269C9515801EDFD761E3E1658D11D50A93CA2305E5674026A586DAAEFF7D71B77FA542C55E48CACE78ED
FD63E36EAE3211B750AEA0813C6961F35E4AED5AC46047C4D53B44E7C1E578C38720CFA21A05D0C907C2DCB87C54E187AD1E2D4291000ECD05FC737EFB1B7B2
E671AD41EF00F118B8E7F58D42D4DC147B93F013C13DFDD5C40C0B22B8F5D4661970B073D9D9CC5B05B4123705A4F8CB72A01DD417C11386E5871BAB6BE6ECA
B0EEB423BA03EA4B96807E4E4D4D36CCFB76709B5A0F3514B9C40A27E734472DAA8D6FA8294B0227C55CEC7691CBFD777F58EEFDEC19582195DE6EBC2BAFA78
3F8101A49C6B825
Time: 00:00:00.0000120
```

```
   Division of 2 numbers using method vs operator:
Method:
Division: 3
Time: 00:00:00.0000116
Operator:
Division: 3
Time: 00:00:00.0002398

   ---------------------------------------------------------------------

   Multiplication of 2 numbers using method vs operator:
Method:
Multiplication: 0869BB3C245F3F1CF4A126637DD36A217CB44F77DC921E2C527A26CA81D3D96C324DD90F2F178F9B6F33A151EF4805A3C2BA6971FBCA56A
D44B37F05012239AF52E8B7EB9F961E38D5DCFD88AF04A6BE05138C642F75DB1A72E01DE2DE730E0A77BCE4B18DA602FE06D767BF024AED92FD4E7CC2CE5803
1C8AB92E267BF6CB7C9FA11238CAC162EDB990D094F3B5A4AC78BB2D30998CF294320186DD4BE1D8B4644FFCEB09AC9EF192F8B82ED8E10D9DFC94E5E1E7C59
F0BC075A2BD995932F713F125DD614EAAC4A0AD93052D611EF3B415E11C54BF38A9FB9F6508B05CAF161C235547E7C318A5BE057E26F08AEADA0C2693C06978
A8902D8CC7EAE80AE81977842B0C99C138E41EEE752CF355398F23E475E3BA97835CCAD6C27E1A3C1A9E0CB8085FA267CAA5A3C5D773AAF18FBE804A818B1EC
D81747DFED1BCA7E5F4BE6EED2F8FC2DAC795A6F31B78FB1D24D4F442CFC8E5A6D074F529633D1F37E60EA99227EE04480F3F994228D550B2368D1CBD4FF8DC
6EF39B4BDEE712D34631BDE153941C91260C5790129C03E477723924AE176398D2BDCF0BF1745015C408013387284586CFE001C7CB4678B8F7E76790EF700BB
72D5A1146DEC10176291A224DE9A87F7074791A42EF865B847F7D8C7FA3C9406D3C4CB196485BA13368F854098871837D42A201DCB9AA5241B916596C669759
6689BF60F5CAF20FE36E
Time: 00:00:00.0000416
Operator:
Multiplication: 0869BB3C245F3F1CF4A126637DD36A217CB44F77DC921E2C527A26CA81D3D96C324DD90F2F178F9B6F33A151EF4805A3C2BA6971FBCA56A
D44B37F05012239AF52E8B7EB9F961E38D5DCFD88AF04A6BE05138C642F75DB1A72E01DE2DE730E0A77BCE4B18DA602FE06D767BF024AED92FD4E7CC2CE5803
1C8AB92E267BF6CB7C9FA11238CAC162EDB990D094F3B5A4AC78BB2D30998CF294320186DD4BE1D8B4644FFCEB09AC9EF192F8B82ED8E10D9DFC94E5E1E7C59
F0BC075A2BD995932F713F125DD614EAAC4A0AD93052D611EF3B415E11C54BF38A9FB9F6508B05CAF161C235547E7C318A5BE057E26F08AEADA0C2693C06978
A8902D8CC7EAE80AE81977842B0C99C138E41EEE752CF355398F23E475E3BA97835CCAD6C27E1A3C1A9E0CB8085FA267CAA5A3C5D773AAF18FBE804A818B1EC
D81747DFED1BCA7E5F4BE6EED2F8FC2DAC795A6F31B78FB1D24D4F442CFC8E5A6D074F529633D1F37E60EA99227EE04480F3F994228D550B2368D1CBD4FF8DC
6EF39B4BDEE712D34631BDE153941C91260C5790129C03E477723924AE176398D2BDCF0BF1745015C408013387284586CFE001C7CB4678B8F7E76790EF700BB
72D5A1146DEC10176291A224DE9A87F7074791A42EF865B847F7D8C7FA3C9406D3C4CB196485BA13368F854098871837D42A201DCB9AA5241B916596C669759
6689BF60F5CAF20FE36E
Time: 00:00:00.0000261
```

```
   Power of number using method vs operator:
Method:
Power: 0BB0588F006E507A0D613A1E1BD95763114CF68D3B7F1CC28FF88C05A75C213A1F640C0BC75DABFA2DBB0E26C532CCC85968DDB29C73945512A6528E
BAE8D53C9E1117F539C74A4BF49D53FD768DF715A640D3293851A7C1610D9157F0EA4DDA15D40CA47075469DD1AD8D9DCD02BD0D45F7F8D6EA0036F2E8ABC74
32F463FD05A89D8AC727DFFBB9F799C11BBFC63F45D76E7A5A967AB0F07761796BA36DEF3836AF4E579D89B64F6E439252A5B98961884D80BFDAE90CC644815
36FB6385462901D4F3253C6B6DA2B419AF93B354869303899FCDDD2E18E79B67519E9F38618BEEDBF67711983D4CB1AD6BF272747A3E3AF69A79F60D6C3B71A
009A7546295A62FC4837FA74ED491529CA74A022867F80FE3BBA652F8D161087D8FD4F9A3DD7F05154BD6D639009B2F32612658C6AC9C436E8FB5388455468F
CB9AECD31766D8A4ED548CAA69900943FE9AAB87FE84D446FBBFC367503E5292ED7C53E75A4C0A3EE931F3B4DB548B5418557AF8CD517EF06FF8482D6E54DD8
B5163FA1AFC4BF27084DF4543315C3BEC5467618B5E90AB9C42E629CFD32D02B48C5F906FADB94BDA2BAAEAA1F7699209EB4CC032CF143FBD3682B820509CD5
948E8F77040CC2595ED6EC4414A603A9F30F457E5133A211C4DA386A2D5220F41DAB905FB6D4B7DE8A6585E1D76389C152A9221111896203C0C970FD2B2C605
3DDB3D8107DCD99B81919B2EAD10D036E6FAE38D582842E2661AAD6480FE53EC8BE8326A5BF26E4A849ED65B14DEB0823D24A4F88B5D084BB652E4D1CCF7286
10D35BFCB239E9CCAA648CE441167E55A68A84AEB1FE4FD6F63278451E68B6D4DB26F60B161D0F688EACE8598FFFD101ED8F157339F8F6356957D92BB63276A
4AB6618267AB58EBBF99FF7DEA29BE46378425EB61232137561589 3C42B9EC328564EB7ABDDCE252FB15C5237B6D80ACA62461A802B9AFD19D23EA2C3591340
186C8E8E33A37EFA14247D22B5060B0043D686DAE8684498045E85B7AE1CE5828D975F7ED68C3AD54E5004403EA16598EA5B8A593E4823C7ED879A95C5FF386
1C6A57855EC68
Time: 00:00:00.0000707
Operator:
Power: 0BB0588F006E507A0D613A1E1BD95763114CF68D3B7F1CC28FF88C05A75C213A1F640C0BC75DABFA2DBB0E26C532CCC85968DDB29C73945512A6528E
BAE8D53C9E1117F539C74A4BF49D53FD768DF715A640D3293851A7C1610D9157F0EA4DDA15D40CA47075469DD1AD8D9DCD02BD0D45F7F8D6EA0036F2E8ABC74
32F463FD05A89D8AC727DFFBB9F799C11BBFC63F45D76E7A5A967AB0F07761796BA36DEF3836AF4E579D89B64F6E439252A5B98961884D80BFDAE90CC644815
36FB6385462901D4F3253C6B6DA2B419AF93B354869303899FCDDD2E18E79B67519E9F38618BEEDBF67711983D4CB1AD6BF272747A3E3AF69A79F60D6C3B71A
009A7546295A62FC4837FA74ED491529CA74A022867F80FE3BBA652F8D161087D8FD4F9A3DD7F05154BD6D639009B2F32612658C6AC9C436E8FB5388455468F
CB9AECD31766D8A4ED548CAA69900943FE9AAB87FE84D446FBBFC367503E5292ED7C53E75A4C0A3EE931F3B4DB548B5418557AF8CD517EF06FF8482D6E54DD8
B5163FA1AFC4BF27084DF4543315C3BEC5467618B5E90AB9C42E629CFD32D02B48C5F906FADB94BDA2BAAEAA1F7699209EB4CC032CF143FBD3682B820509CD5
948E8F77040CC2595ED6EC4414A603A9F30F457E5133A211C4DA386A2D5220F41DAB905FB6D4B7DE8A6585E1D76389C152A9221111896203C0C970FD2B2C605
3DDB3D8107DCD99B81919B2EAD10D036E6FAE38D582842E2661AAD6480FE53EC8BE8326A5BF26E4A849ED65B14DEB0823D24A4F88B5D084BB652E4D1CCF7286
10D35BFCB239E9CCAA648CE441167E55A68A84AEB1FE4FD6F63278451E68B6D4DB26F60B161D0F688EACE8598FFFD101ED8F157339F8F6356957D92BB63276A
4AB6618267AB58EBBF99FF7DEA29BE46378425EB61232137561589 3C42B9EC328564EB7ABDDCE252FB15C5237B6D80ACA62461A802B9AFD19D23EA2C3591340
186C8E8E33A37EFA14247D22B5060B0043D686DAE8684498045E85B7AE1CE5828D975F7ED68C3AD54E5004403EA16598EA5B8A593E4823C7ED879A95C5FF386
1C6A57855EC68
Time: 00:00:00.0000397
```

Висновок:

Було досліджено алгоритми реалізації арифметичних операцій над великими (багато розрядними) числами над скінченими полями та групами з точки зору їх ефективності за часом на мові C#. Ми навчилися працювати з класом BigInteger, його методами, конструкторами та операторами, створили свої власні методи для зручної роботи з цим класом. Також порівняли виконання однакових арифметичних операцій використовуючи методи та оператори цього касу на однакових числах, можемо зробити висновок, що оператори в середньому працюють швидше.