## СПЕЦІАЛЬНІ РОЗДІЛИ

## ОБЧИСЛЮВАЛЬНОЇ МАТЕМАТИКИ

## КОМП’ЮТЕРНИЙ ПРАКТИКУМ No1

## Багаторозрядна арифметика

1. Мета роботи

Отримання практичних навичок програмної реалізації багаторозрядної арифметики; ознайомлення з прийомами ефективної реалізації критичних по часу ділянок програмного коду та методами оцінки їх ефективності.

2. Завдання до комп’ютерного практикуму

А) Згідно варіанту розробити клас чи бібліотеку функцій для роботи з m-бітними цілими числами. Бібліотека повинна підтримувати числа довжини до 2048 біт.

Повинні бути реалізовані такі операції:

1) переведення малих констант у формат великого числа (зокрема, 0 та 1);

2) додавання чисел;

3) віднімання чисел;

4) множення чисел, піднесення чисел до квадрату;

5) ділення чисел, знаходження остачі від ділення;

6) піднесення числа до багаторозрядного степеня;

7) конвертування (переведення) числа в символьну строку та обернене перетворення символьної строки у число; обов’язкова підтримка шістнадцяткового представлення, бажана – десяткового та двійкового.

Код програми виконаний мовою JavaScript

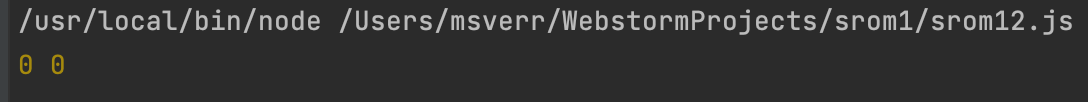
Перевірка коректності:

let ***s1*** = hexMul(hexSum(***First***.hex,***Second***.hex),***Third***.hex);

let ***s2*** = hexMul(***Third***.hex,hexSum(***First***.hex,***Second***.hex));

let ***s3*** = hexSum(hexMul(***First***.hex,***Third***.hex),hexMul(***Second***.hex,***Third***.hex));

***console***.log(compare(***s1***,***s2***), compare(***s2***,***s3***));



Функція compare виводить 0 коли на вхід подаються однакові значення, з чого робимо висновок що s1=s2=s3, отже операції коректні

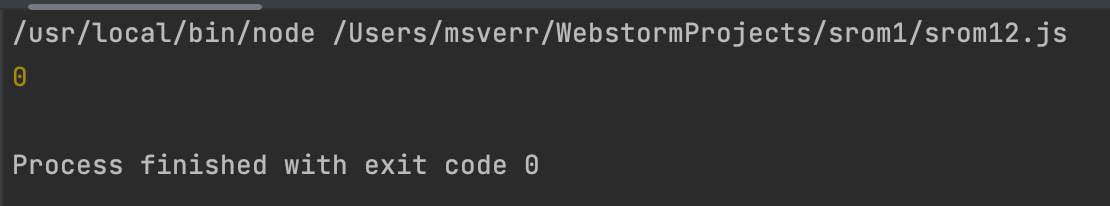
let ***result*** = hexMul(***First***,"100"), ***res*** = "";

for(let i = 0; i < 100; i++){

***res*** = hexSum(***res***,***First***);

}

***console***.log(compare(***result***,***res***));



З чого робимо висновок про коректність множення на константу

Час виконання операцій (взято середнє значення при 10 запусках)

| Операція/значення | Мінімальне | Максимальне | Середнє |
| --- | --- | --- | --- |
| Додавання | 0.295ms | 0.986ms | 0.63ms |
| Віднімання | 0.273ms | 5.518ms | 1.27ms |
| Множення | 40.411ms | 83.316ms | 48.77ms |
| Піднесення до степеня | 2.929s | 3.046s | 2.97s |
| Ділення | 0.564ms | 7.452ms | 2.11ms |

// Степінь використовувався чотиризначний

// Ділення відбувалось на значення, приблизно схоже на значення дільника, якщо 256-значний hex ділити на чотиризначний, виконається 250+ тисяч операцій, не бачу сенсу перевіряти такий випадок через нешвидкодію виконання такої кількості операцій