

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 6 з
дисципліни «Алгоритми та структури
даних-1. Основи алгоритмізації»

«Дослідження рекурсивних алгоритмів»

Варіант 30

Виконав студент ІІ-13 Симолюк Денис Андрійович
(шифр, прізвище, ім'я, по батькові)

Перевірів Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Київ 20211

Лабораторна робота 6

Дослідження рекурсивних алгоритмів

Мета – дослідити особливості роботи рекурсивних алгоритмів та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм.

Варіант 30

Дано перший член і знаменник геометричної прогресії. Обчислити суму n перших членів прогресії та знайти n -й член прогресії.

Постановка задачі

Для вирішення задачі необхідно створити рекурсивну функцію, яка буде обраховувати n -й член прогресії. Сума ж перших n членів буде обчислюватися за формулою знаходження суми геометричної прогресії. Також потрібно перевірити чи q дорівнює 1. Якщо так, користувачу запропонується ввести дані ще раз.

Побудова математичної моделі

Так як ми працюємо з геометричною прогресією, то кожен наступний її член (b_n) дорівнює попередньому, помноженому на знаменник (q) – число, відмінне від одиниці:

$$b_n = b_{n-1} * q$$

Суми ми знаходимо за формулами сум геометричної прогресії залежно від значення q :

$$S_n = \frac{b_1(q^n - 1)}{q - 1}, q \neq 1$$

$$S = \frac{b_1}{1 - q}, |q| < 1$$

Побудуємо таблицю змінних:

Змінна	Тип	Призначення
Даний перший член прогресії b_1	Дійсна	Вхідні дані
Даний знаменник прогресії q	Дійсна	Вхідні дані
Номер n -го члена n	Ціла	Вхідні дані
Чисельник в сумі членів прогресії num	Дійсна	Проміжні дані
Знаменник в сумі den	Дійсна	Проміжні дані
Шукана сума членів прогресії sum	Дійсна	Вихідні дані
Шуканий n -й член b_n	Дійсна	Вихідні дані

Крок 1. Визначимо основні дії.

Крок 2. Перевіримо q .

Крок 3. Деталізуємо знаходження sum .

Крок 4. Деталізуємо знаходження num .

Крок 5. Деталізуємо знаходження den .

Крок 6. Деталізуємо обчислення sum .

Крок 7. Деталізуємо виклик рекурсивної функції.

Крок 8. Деталізуємо обчислення b_n .

Псевдокод алгоритму

Крок 1.

початок

введення b_1, q, n

перевірка q

знаходження sum

рекурсивна функція

виведення sum, bn

кінець

Крок 2.

початок

введення b1, q, n

поки $q == 1$

повторити

введення q

все повторити

знаходження sum

рекурсивна функція

виведення sum, bn

кінець

Крок 3.

початок

введення b1, q, n

поки $q == 1$

повторити

введення q

все повторити

якщо $\text{abs}(q) > 1$

то

обчислення num

обчислення den

інакше

обчислення num

обчислення den

все якщо

рекурсивна функція

виведення sum, bn

кінець

Крок 4.

початок

введення b1, q, n

поки $q == 1$

повторити

введення q

все повторити

якщо $\text{abs}(q) > 1$

то

$\text{num} = b1 * (\text{pow}(q, n) - 1)$

обчислення den

інакше

$\text{num} = b1$

обчислення den

все якщо

рекурсивна функція

виведення sum, bn

кінець

Крок 5.

початок

введення b1, q, n

поки $q == 1$

повторити

введення q

все повторити

якщо $\text{abs}(q) > 1$

то

$\text{num} = b1 * (\text{pow}(q, n) - 1)$

$\text{den} = q - 1$

інакше

$\text{num} = b1$

$\text{den} = 1 - q$

все якщо

обчислення sum

рекурсивна функція

виведення sum, bn

кінець

Крок 6.

початок

введення b1, q, n

поки $q == 1$

повторити

введення q

все повторити

якщо $\text{abs}(q) > 1$

то

$\text{num} = b1 * (\text{pow}(q, n) - 1)$

$\text{den} = q - 1$

інакше

$\text{num} = b1$

$\text{den} = 1 - q$

все якщо

$sum = num / den$

рекурсивна функція

виведення sum, bn

кінець

Крок 7.

початок

введення $b1, q, n$

поки $q == 1$

повторити

введення q

все повторити

якщо $abs(q) > 1$

то

$num = b1 * (pow(q, n) - 1)$

$den = q - 1$

інакше

$num = b1$

$den = 1 - q$

все якщо

$sum = num / den$

$bn = countbn(b1, q, n)$

виведення sum, bn

кінець

підпрограма countbn (b, step, quantity)

початок

обчислення bn

кінець

Крок 8.

початок

введення $b1, q, n$

поки $q == 1$

повторити

введення q

все повторити

якщо $\text{abs}(q) > 1$

то

$\text{num} = b1 * (\text{pow}(q, n) - 1)$

$\text{den} = q - 1$

інакше

$\text{num} = b1$

$\text{den} = 1 - q$

все якщо

$\text{sum} = \text{num} / \text{den}$

$b_n = \text{countbn}(b1, q, n)$

виведення sum, b_n

кінець

підпрограма countbn (b, step, quantity)

початок

якщо $\text{quantity} != 1$

то

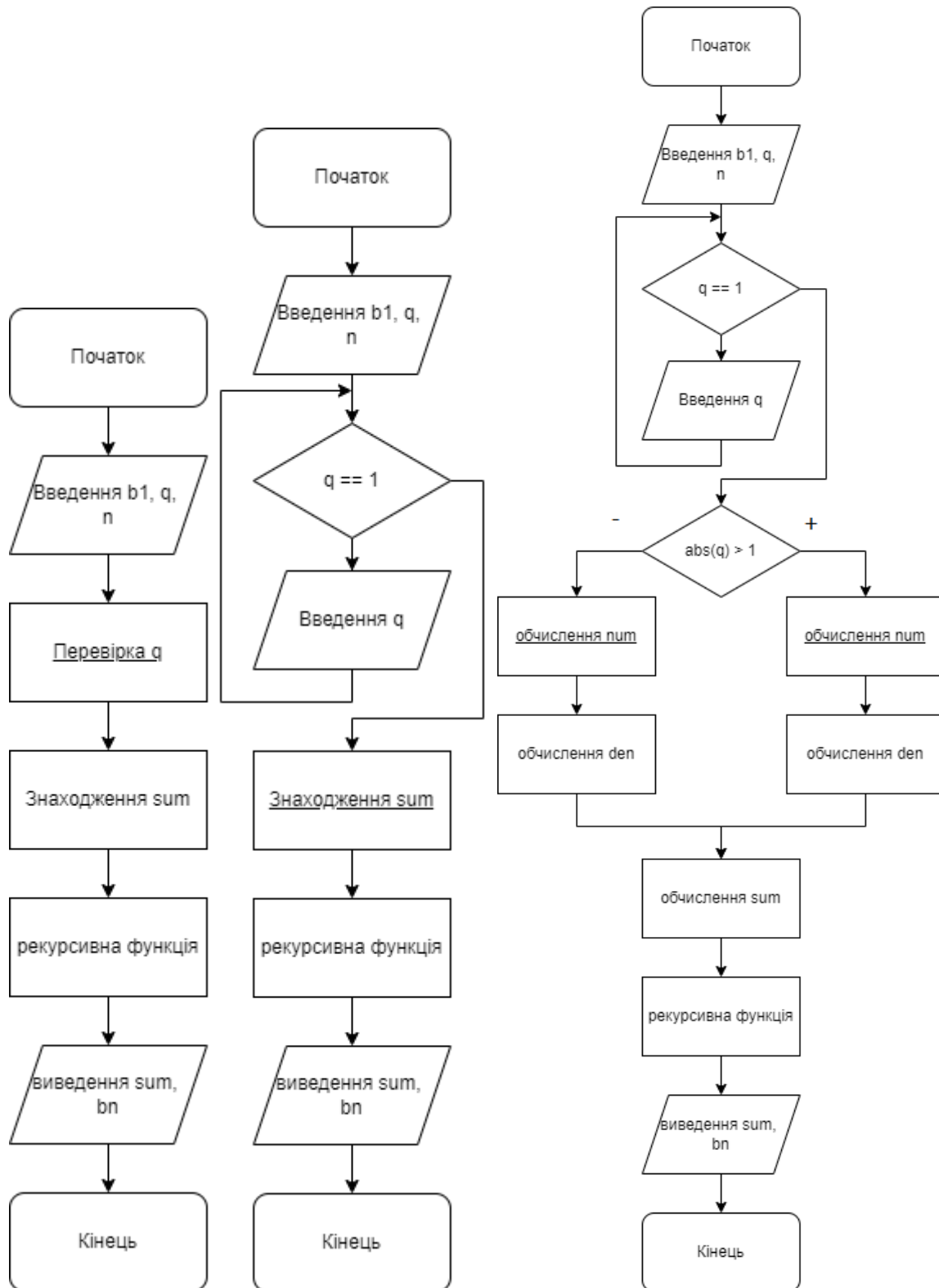
$b = \text{countbn}(b, \text{step}, \text{quantity} - 1) * \text{step}$

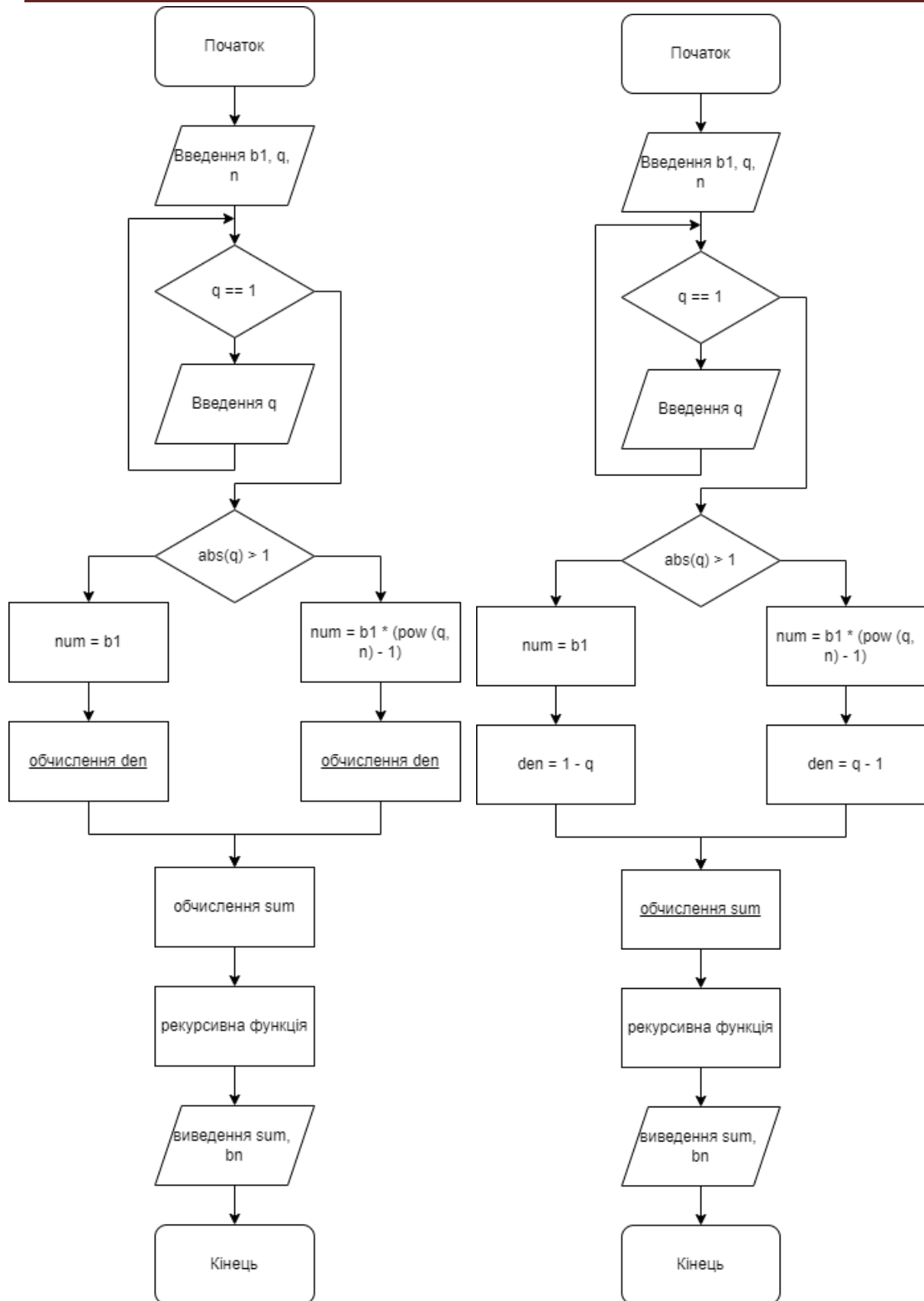
все якщо

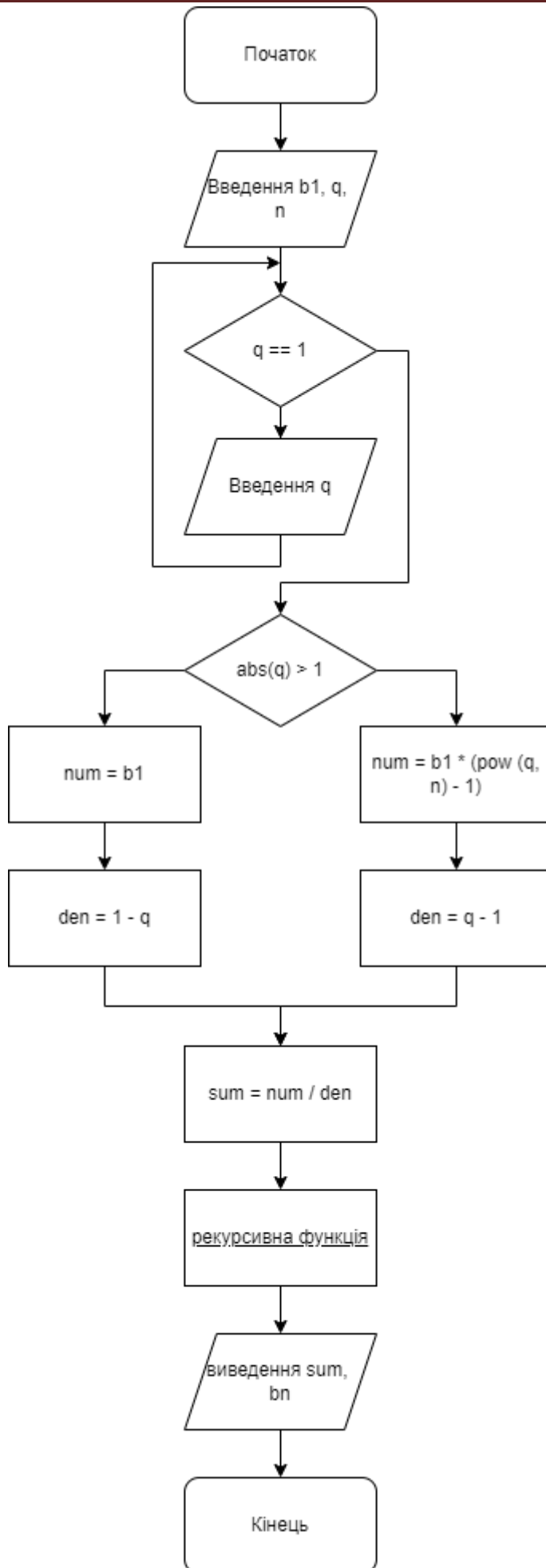
повернення b

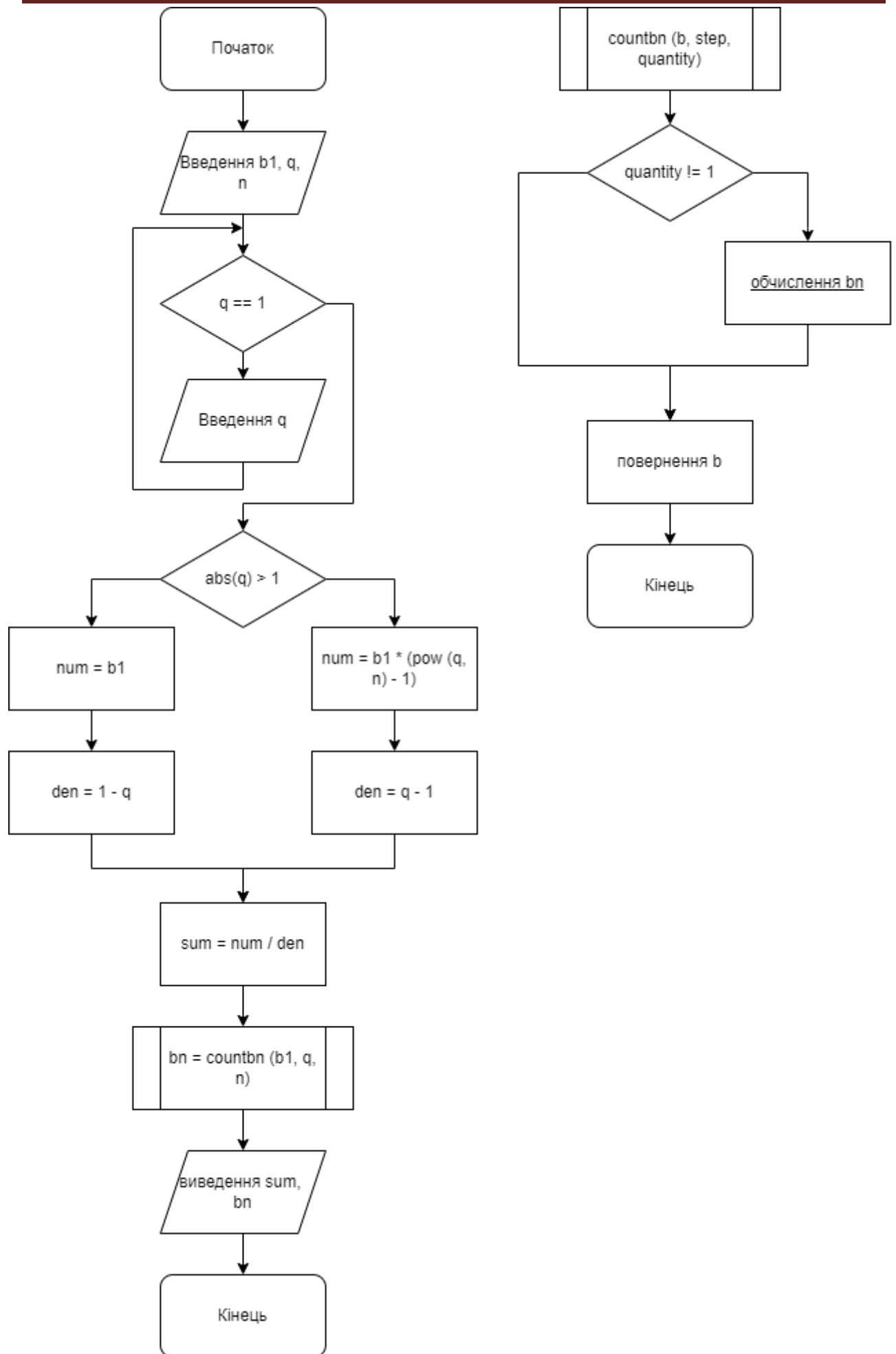
кінець

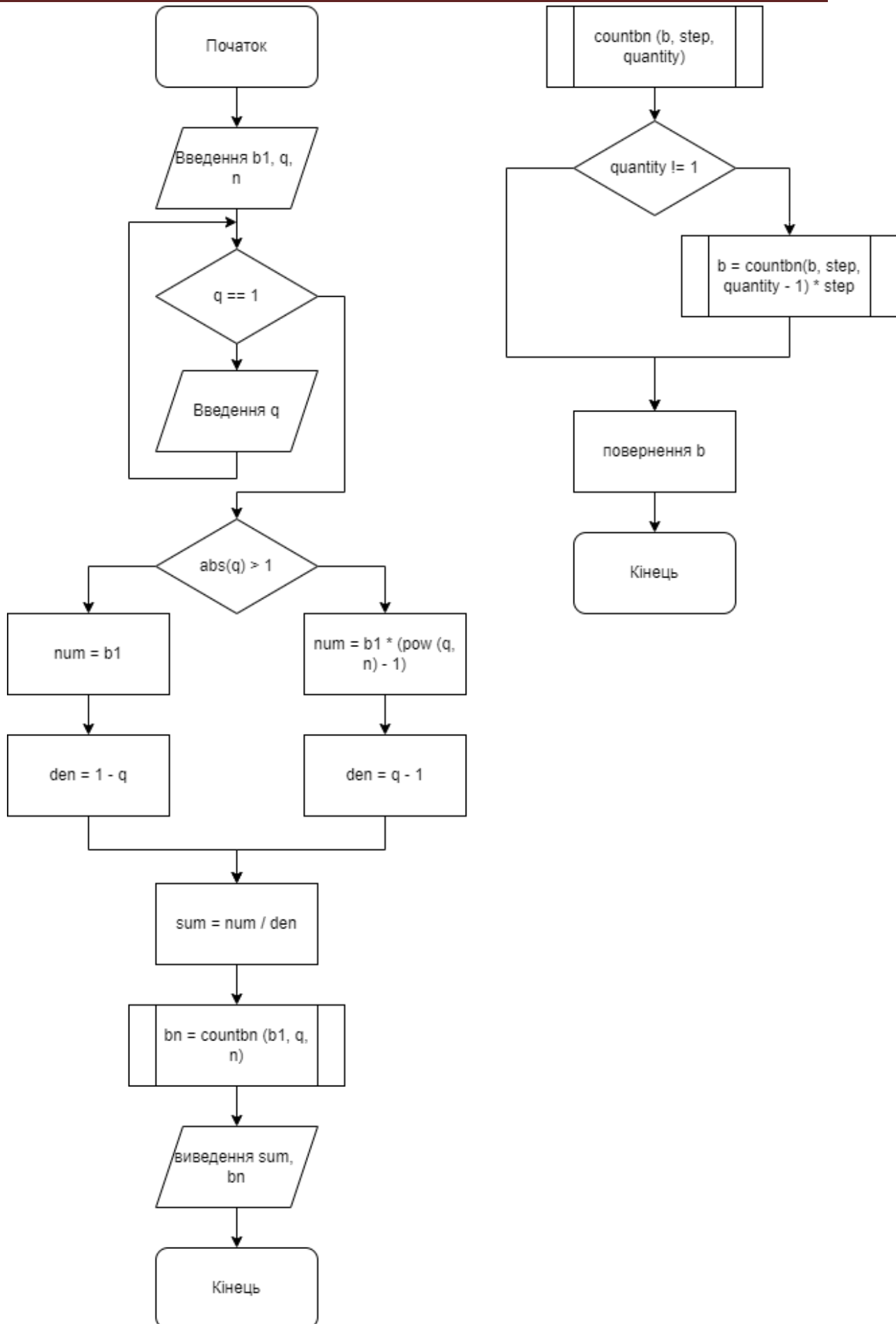
Блок-схема алгоритму











Код алгоритму

```

1
2     #include <iostream>
3     using namespace std;
4     double countbn(double, double, int);
5
6     int main()
7     {
8         double b1, bn, q, sum, den, num;
9         int n;
10        cout << "b1 = ";
11        cin >> b1;
12        cout << "q = ";
13        cin >> q;
14        while (q == 1)
15        {
16            cout << "invalid item. Try again" << endl;
17            cout << "q = ";
18            cin >> q;
19        }
20        cout << "n = ";
21        cin >> n;
22        if (abs(q) > 1)
23        {
24            num = b1 * (pow(q, n) - 1);
25            den = q - 1;
26        }
27        else
28        {
29            num = b1;
30            den = 1 - q;
31        }
32        sum = num / den;
33        bn = countbn(b1, q, n);
34        cout << "sum = " << sum << endl;
35        cout << "b" << n << " = " << bn;
36    }
37
38    double countbn(double b, double step, int quantity)
39    {
40        if (quantity != 1)
41        {
42            b = countbn(b, step, quantity - 1) * step;
43        }
44        return b;
45    }
46

```

Консоль отладки Microsoft Visual Studio

b1 = 2
q = 2
n = 5
sum = 62
b5 = 32

D:\Projects\ab 6 asd\Debug\ab 6 asd.exe (процесс 13980) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно...

Випробування алгоритму

Блок	Дія
1	Введення $b1 = 2, q = 2, n = 5$
2	$q \neq 1$
3	Розгалуження $q > 1$
4	$num = b1 * (pow(q, n) - 1)$
5	$den = q - 1$
6	$sum = num / den$
7	$bn = countbn(2, 2, 5)$
1	$b = countbn(2, 2, 4) * 2$
2	$b = countbn(2, 2, 3) * 2$
3	$b = countbn(2, 2, 2) * 2$
4	$b = countbn(2, 2, 1) * 2$ повернення b
5	$countbn(2, 2, 1) = 2$ $b = 2 * 2 = 4$ повернення b
6	$countbn(2, 2, 2) = 4$ $b = 4 * 2 = 8$ повернення b
7	$countbn(2, 2, 3) = 8$ $b = 8 * 2 = 16$ повернення b
8	$countbn(2, 2, 4) = 16$ $b = 16 * 2 = 32$ повернення b
8	$bn = 32$
6	Виведення sum, bn

Висновок

В даній роботі я дослідив особливості роботи рекурсивних алгоритмів та набув практичних навичок їх використання під час складання програмних специфікацій підпрограм, зокрема під час роботи з мовою програмування C++.