

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 № 5 з дисципліни

«Алгоритми та структури даних

1. Основи алгоритмізації»

«Організація циклічних процесів. Складні цикли»

Варіант 22

Виконав студент ІП-14 Нікулін Павло Юрійович
(шифр, прізвище, ім'я, по батькові)

Перевірів Мартінова Оксана Петрівна
(прізвище, ім'я, по батькові)

Лабораторна робота 5

Організація циклічних процесів. Складні цикли

Мета: вивчити особливості організації складних циклів.

Завдання

Натуральне число називається паліндромом, якщо його запис читається однаково з початку та з кінця. Знайти всі паліндроми з інтервалу [1 000 000, 1 000 000 000].

Розв'язання

1. **Постановка задачі:** результатом роботи має бути всі натуральні числа, що є паліндромами на заданому проміжку. Для пошуку паліндромів потрібно «перегорнути» число n . Якщо воно буде дорівнювати числу n , то це паліндром.
2. Побудова **математичної моделі**. Складемо таблицю імен змінних.

Змінна	Тип	Ім'я	Призначення
Число з проміжку	Натуральне	N	Початкове дане
Число для операцій	Натуральне	TEMP	Проміжне дане
Цифра числа з проміжку	Натуральне	NUM	Проміжне дане
Реверсія числа з проміжку	Натуральне	REV	Проміжне дане

Крок 1. Визначимо основні дії.

Крок 2. Позначимо проміжок числа n .

Крок 3. Знайдемо реверсію числа.

Крок 4. Порівняємо число з його реверсією.

Псевдокод

крок 1

початок

Позначимо проміжок числа n

Знайдемо реверсію числа

Порівняємо число з його реверсією

Кінець

крок 2

початок

$n = 1000000$

повторити

Знайдемо реверсію числа

$n += 1$

поки $n \leq 1000000000$

все повторити

Порівняємо число з його реверсією

кінець

крок 3

початок

$n = 1000000$

повторити

$temp = n, rev = 0$

повторити

$num = temp \% 10$

$rev = rev * 10 + num$

```

    temp = temp // 10

    поки temp < 0

    все повторити

    n += 1

поки n <= 1000000000

все повторити

Порівняємо число з його реверсією

кінець

крок 4

початок

n = 1000000

повторити

    повторити

        num = temp % 10

        rev = rev * 10 + num

        temp = temp // 10

    поки temp > 0

    все повторити

    n += 1

поки n <= 1000000000

все повторити

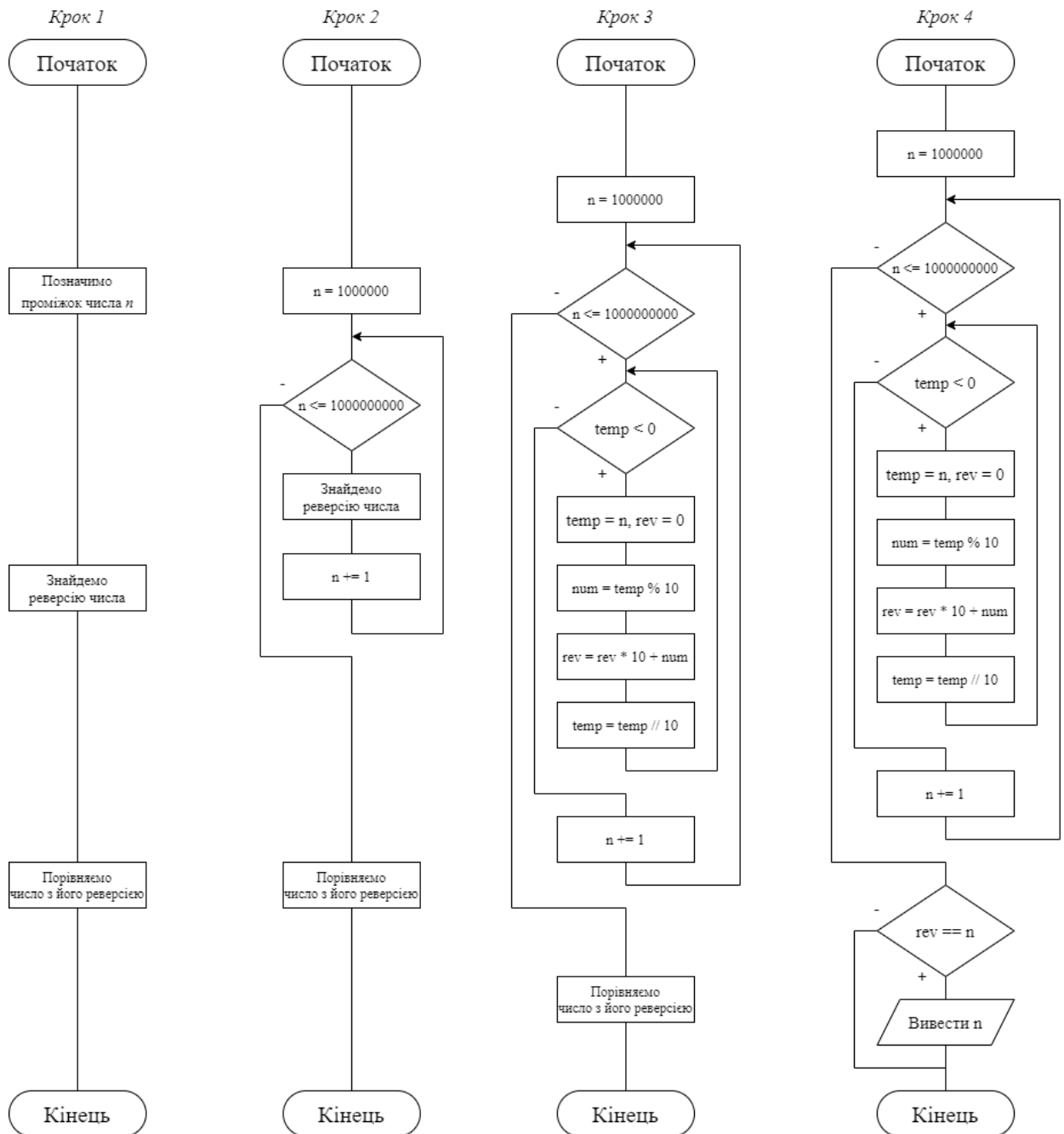
якщо rev == n то

    print(n)

кінець

```

Блок-схема



Випробування алгоритму

Початок

```
n = 1000000
```

```
1 while n <= 10000000000:
```

```
    temp = n
```

```
    rev = 0
```

```
2     while temp > 0:
```

```
        num = temp % 10
```

```
        rev = rev * 10 + num
```

```
        temp = temp // 10
```

```
3     if rev == n:
```

```
        print(str(n))
```

```
        n += 1
```

Кінець

Висновок

Під час виконання лабораторної роботи було досліджено особливості організації складних циклів. Було використано два цикли *while*: для задання інтервалу числа *x* та для «перегортання» цього ж числа. Також використано умовний оператор *if* для порівняння числа *x* та його дзеркального відображення. Використана структура вкладеного циклу з передумовою.