**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

«Основи програмування

1. Базові конструкції»

«Організація циклічних процесів. Складні цикли»

Варіант 22

Виконав студент ІП-14 Нікулін Павло Юрійович

(шифр, прізвище, ім'я, по батькові)

Перевірив Новікова Поліна Анатоліївна

(прізвище, ім'я, по батькові)

**Лабораторна робота №5**

**Організація циклічних процесів. Складні цикли**

**Мета**: вивчити особливості організації складних циклів.

**Хід роботи**

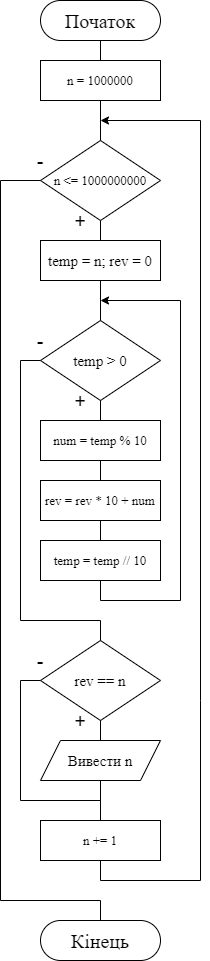
**Задача**: натуральне число називається паліндромом, якщо його запис читається однаково з початку та з кінця. Знайти всі паліндроми з інтервалу [1 000 000, 1 000 000 000].

*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 |

*Блок-схема*

****

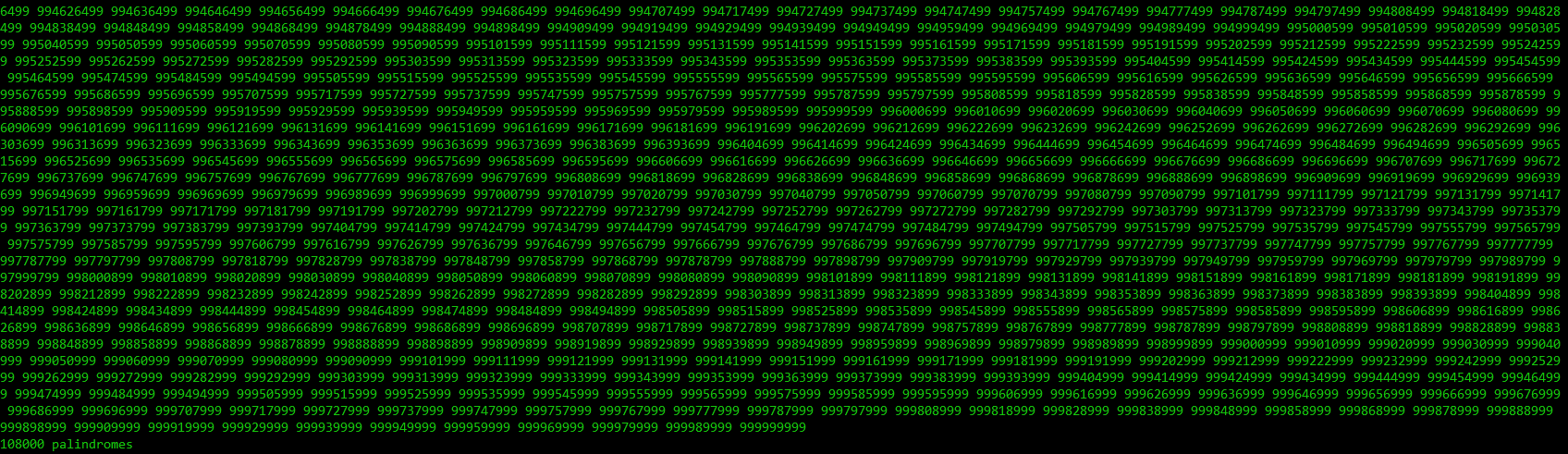
**Випробування коду**

**Python**

*Код*

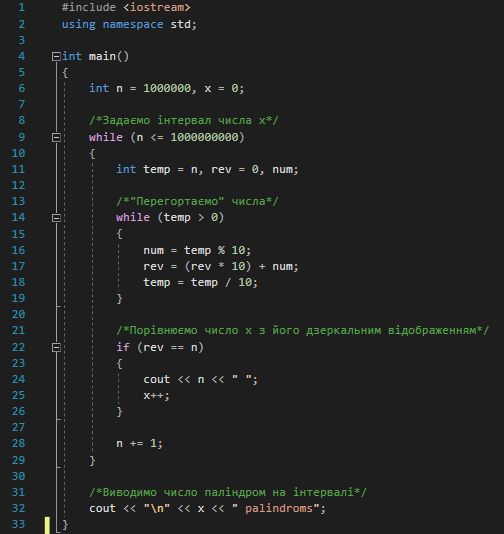
****

*Результат (весь результат не вміщається на скріншоті)*

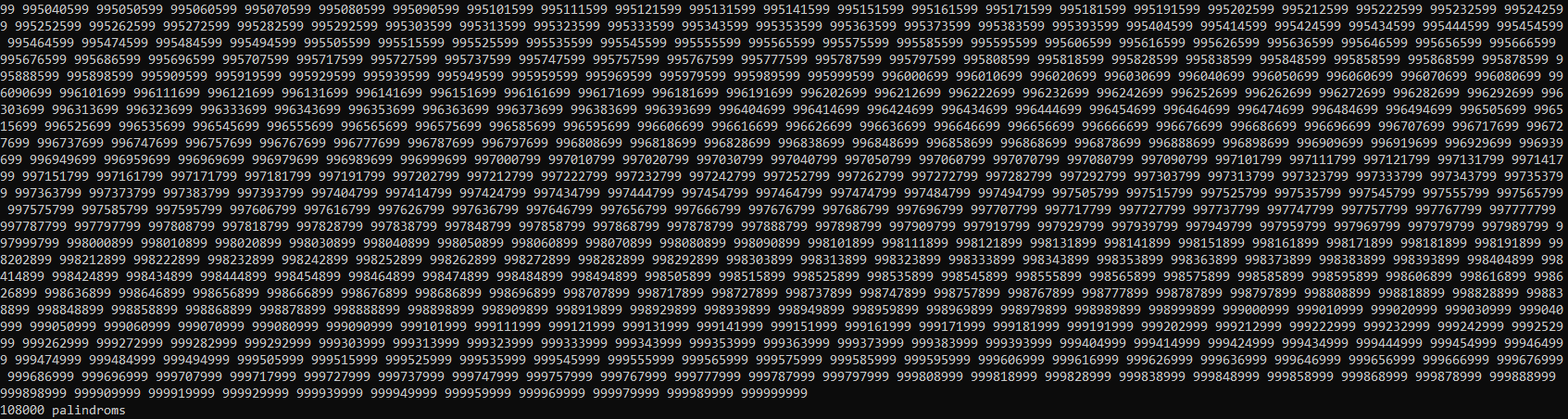
****

**C++**

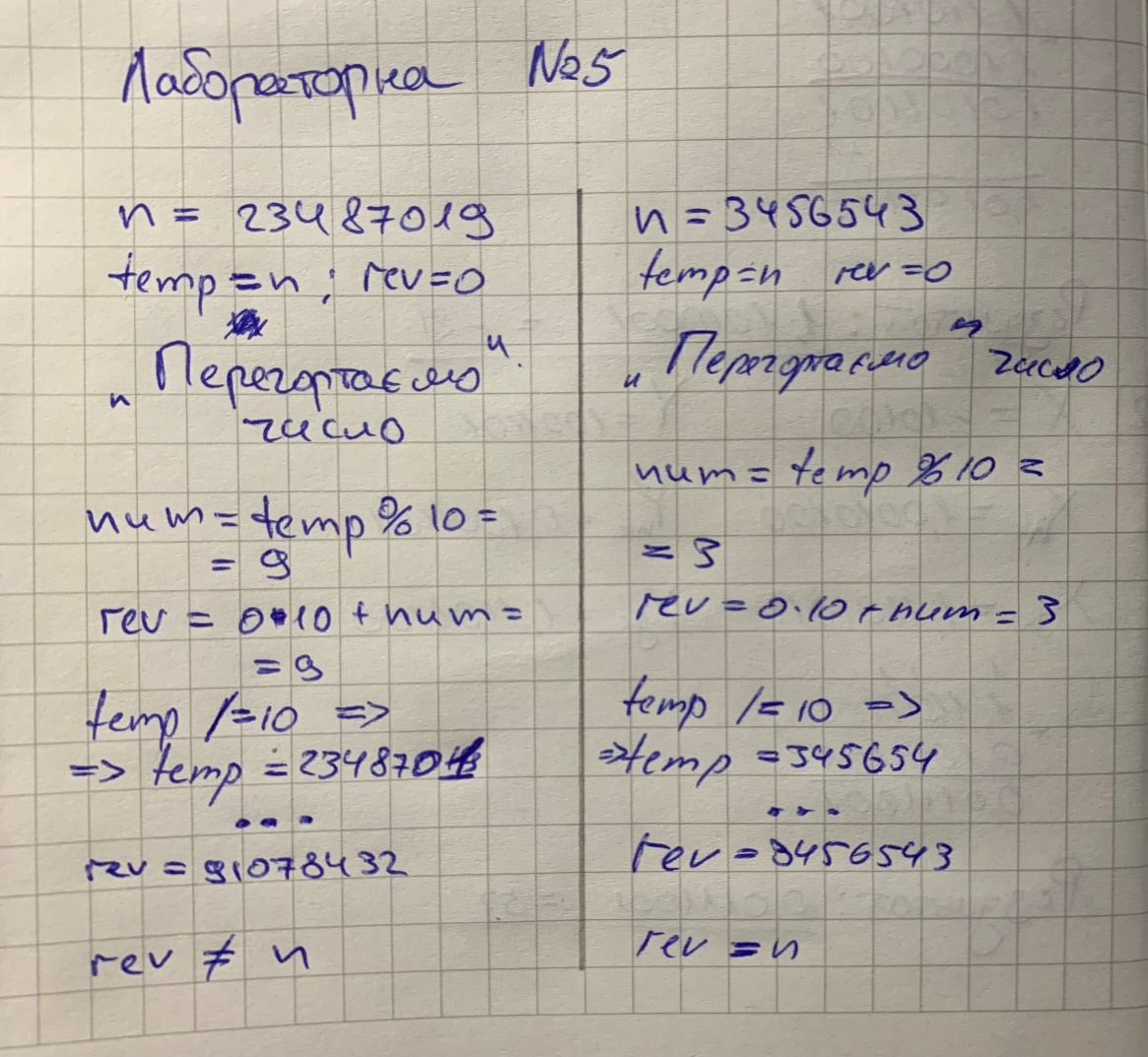
*Код*

****

*Результат (весь результат не вміщається на скріншоті)*



*Математичне доведення*

****

**Висновок**

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