

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 27

Виконав студент ІП-11 Савенко Олексій Андрійович

(шифр, прізвище, ім'я, по батькові)

Перевірів Вітковська І.І.

(прізвище, ім'я, по батькові)

Київ 2021

## Лабораторна робота 5

### Організація циклічних процесів Складні цикли

**Мета роботи** – вивчити особливості організації складних циклів.

#### Варіант 27

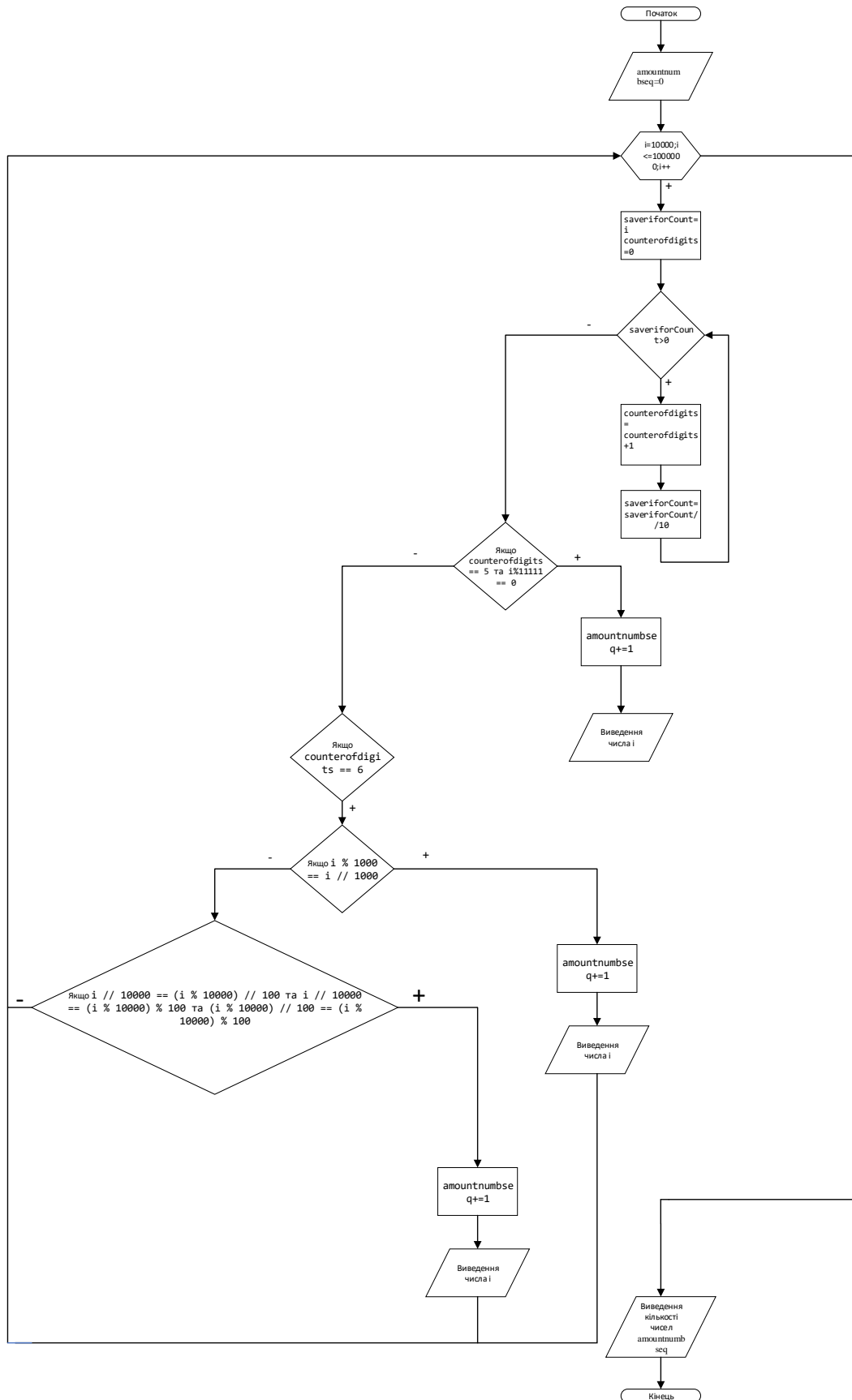
##### Індивідуальне завдання

27. Знайти всі числа, що представляють собою повторювані фрагменти послідовності цифр (3434, 23452345) з інтервалу [10000, 1000000].

##### Постановка завдання

Результат завдання є всі числа виду  $n_1n_2n_3...n_1n_2n_3...$  з проміжку [10000, 1000000], їх відбір здійснюється за допомогою арифметичного циклу **for**, в залежності від кількості цифр у відповідному числі, кількість цифр ми отримаємо за допомогою ітераційного циклу **while** вкладеного в арифметичний цикл, в залежності від кількості цифр 5 або 6 виконується порівняння частин числа, якщо умова виконується число підходить за умовою завдання, отже потрібно вивести його значення. Після кожного виведення числа збільшується значення змінної **amountnumbseq** на 1, після завершення перебирання чисел з проміжку вказаному у завданні, а отже виходу з арифметичного циклу **for** відбувається виведення кількості елементів, що представляють собою повторювані фрагменти послідовності цифр.

Блок схема алгоритму:



### Код програми на мові програмування C++:

```
#include <iostream>
using namespace std;

int main() {
    setlocale(LC_ALL, "Russian");
    int saveriforCount, amountnumbseq = 0;
    int firstnumber = 10000;
    int lastnumber = 1000000;

    for (int i = firstnumber; i <= lastnumber; i++) {
        saveriforCount = i;
        int counterofdigits = 0;
        while (saveriforCount > 0) {
            counterofdigits++;
            saveriforCount = saveriforCount / 10;
        }

        if (counterofdigits == 5 && i % 11111 == 0) {
            amountnumbseq++;
            cout << i << "\n";
        }
        else if (counterofdigits == 6) {

            if (i % 1000 == i / 1000) {
                amountnumbseq++;
                cout << i << "\n";
            }

            else if (i / 10000 == (i % 10000) / 100 && i / 10000 ==
(i % 10000) % 100 && (i % 10000) / 100 == (i % 10000) % 100) {
                amountnumbseq++;
                cout << i << "\n";
            }
        }
    }
}
```

```

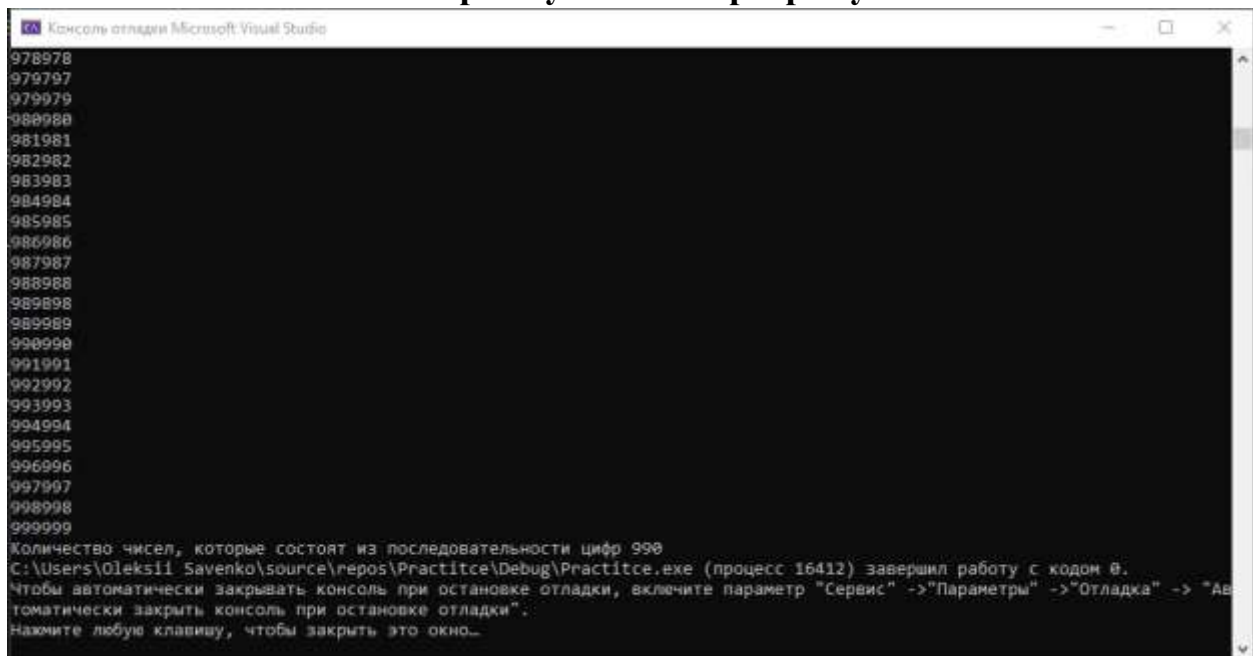
    }

    }

    cout << "Количество чисел, которые состоят из последовательности
цифр " << amountnumbseq;
    return 0;
}

```

### Виконання алгоритму на мові програмування C++:



```

Консоль отладки Microsoft Visual Studio
978978
979797
979979
980980
981981
982982
983983
984984
985985
986986
987987
988988
989989
990990
991991
992992
993993
994994
995995
996996
997997
998998
999999
Количество чисел, которые состоят из последовательности цифр 990
C:\Users\Oleksii.Savenko\source\repos\Practitce\Debug\Practitce.exe (процесс 16412) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно.

```

### Код програми на мові програмування Python:

```
amountnumbseq=0
```

```

for i in range(10000,1000000):
    saveriforCount=i
    counterofdigits=0
    while saveriforCount>0:
        counterofdigits+=1
        saveriforCount=saveriforCount//10

    if counterofdigits == 5 and i%11111 == 0:
        amountnumbseq+=1
        print(i)

    elif counterofdigits == 6:

        if i % 1000 == i // 1000 :
            amountnumbseq+=1
            print(i)

        elif i // 10000 == (i % 10000) // 100 and i // 10000 == (i % 10000) % 100
and (i % 10000) // 100 == (i % 10000) % 100:
            amountnumbseq+=1
            print(i)

print("Количество чисел,которые состоят из последовательности цифр " +
str(amountnumbseq))

```

## Виконання алгоритму на мові програмування Python:



```
C:\Users\Oleksii Savenkiv\AppData\Local\Programs\Python\Python39\python.exe
974974
975975
976976
977977
978978
979797
979979
980980
981981
982982
983983
984984
985985
986986
987987
988988
989989
989989
990990
991991
992992
993993
994994
995995
996996
997997
998998
999999
Количество чисел, которые состоят из последовательности цифр 990
Press any key to continue . . .
```

## Висновок

Таким чином, я вивчив особливості організації складних циклів. Я створив алгоритм для відбору та виведення чисел, які являють собою повторювані фрагменти послідовності цифр, основою для нього став арифметичний цикл **for**, головне завдання якого перебір чисел у проміжку від 10000 до 1000000, із кроком **i++**, для визначення кількості цифр у числі був використаний вкладений ітераційний цикл **while**. Для перевірки чисел за умовою завдання був використаний оператор умови **if** та поєднаний **else if** для доповненої умови у разі невиконання першої. Після виконання арифметичного циклу **for**, було виведено кількість відповідних чисел – **990**. Отже, мною були написані алгоритм та програмний код на його основі, за допомогою якого він був протестований та відповідно була доведена його вірність.