

Лабораторна робота № 5.

Тема заняття: Цикли з передумовою та післяумовою (**while** та **do while**).

Мета: Одержання практичних навичок у роботі з циклічними обчисленнями та циклічною обробкою даних у програмах мови C++. Відпрацювання використання циклів з передумовою та післяумовою.

Зміст роботи.

Завдання 1. Написати програму для рішення задач згідно варіанту. Програми написати у двох варіантах використовуючи цикли **while** та **do while**.

Варіант 1.

1. Скільки цифр даного числа не кратні N? Чи правильно, що в даному числі сума всіх цифр дорівнює K?
2. Дано натуральне число. Приписати до нього таке саме натуральне число в зворотному порядку.

Варіант 2.

1. Знайти суму цифр числа. Чи правильно, що число починається і закінчується однією цифрою?
2. Із даного натурального числа вилучити всі нулі.

Варіант 3.

1. Знайти добуток цифр числа. Чи правильно, що в даному числі відсутня цифра n?
2. Знайти всі прості дільники даного числа.

Варіант 4.

1. Підрахувати кількість цифр даного числа. Чи правильно, що дана цифра зустрічається більше двох разів.
2. Дано число. З даного числа знайшли суму цифр. З утвореного числа знов знайшли суму цифр, і.т.д. поки не утворилась одна цифра.

Варіант 5.

1. Знайти кількість парних цифр даного числа. Чи правильно, що сума цифр даного числа більша за S?
2. Знайти кількість непарних дільників даного числа.

Варіант 6.

1. Знайти другу цифру в даному числі. Чи правильно, що сума парних цифр дорівнює N?
2. Дано натуральне число $N > 0$. Від числа відняти суму цифр цього числа, від утвореного числа знов віднімають суму цифр утвореного числа. Дану операцію виконують поки утворене число додатне. Скільки разів буде виконуватись дана операція?

Варіант 7.

1. Знайти кількість цифр даного числа, які більші за n. Чи правильно, що число складається тільки з парних цифр?
2. Знайти кількість простих дільників даного числа.

Варіант 8.

1. Знайти першу і останню цифри даного числа. Чи правильно, що в числі відсутні нулі?
2. Дано натуральне число Z приписати до нього в кінець першу і останню цифри цього числа.

Варіант 9.

1. Переверніть задане число зліва на право (2342 $\rightarrow$ 2432). Чи правильно, що сума цифр даного числа кратна N?
2. Знайти кількість парних дільників даного числа.

Варіант 10.

1. Скільки цифр даного числа кратні N ? Чи правильно, що в даному числі сума першої і останньої цифр дорівнює K ?
2. Дано натуральне число. Приписати до нього таке саме натуральне число.

Варіант 11.

1. Підрахувати, скільки разів дана цифра зустрічається в даному числі. Чи правильно, що дане число паліндром?
2. Дано дріб m/n . Скоротити його і записати результат у вигляді дробу.

Варіант 12.

1. Скільки цифр в даному числі, які не дорівнюють A ? Чи правильно що в числі всі цифри однакові?
2. Із даного натурального числа вилучити всі цифри A

Варіант 13.

1. Скільки цифр в даному числі, які дорівнюють A ? Чи правильно що в числі всі цифри різні?
2. Знайти найменше спільне кратне двох натуральних чисел.

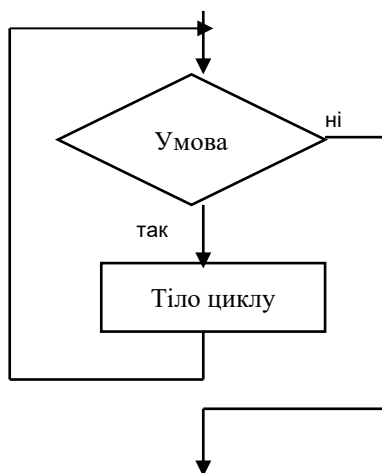
Варіант 14.

1. Скільки цифр в даному числі, які кратні передостанній цифрі? Чи правильно що в числі дві перші цифри однакові?
2. Знайти всі симетричні числа на проміжку від a до b .

Варіант 15.

1. Скільки цифр в даному числі, які не дорівнюють A ? Чи правильно що в числі всі цифри рівні?
2. Розкласти натуральне число на прості множники.

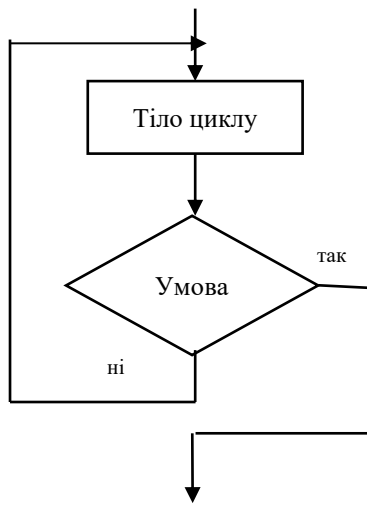
Методичні рекомендації



Цикл з передумовою – цикл, який виконується поки істинна деяка умова.

ПОКИ логічний вираз істинний ВИКОНУВАТИ оператор(и).

```
while (умова) {  
    тіло циклу;  
}
```



Цикл з післяумовою – цикл, який виконується поки деяка умова не стане істиною.

ПОВТОРЮВАТИ оператор(и) ДОТИ, ПОКИ логічний вираз не стане істинним (TRUE).

```
do {
    тіло циклу;
} while (умова)
```

Синтаксис while у загальному вигляді наступний:

```
while (умова)
    оператор
```

Умова циклу – як і в умовному операторі if – це вираз, що приймає логічне значення "істина" або "фальш". Виконання оператора повторюється доти, доки значенням умови є true (істина). Умова обчислюється заново перед кожною ітерацією.

У цьому записі під елементом оператор розуміємо або одиночний оператор, або блок операторів. Елемент оператор здійснюється доти, доки умовний вираз повертає значення ІСТИНА. Як тільки цей вираз стає помиловим, то керування передається оператору, який знаходиться поза цим циклом.

Приклад 1. Обчислити суму цілих чисел від 1 до 1000.

```
int main()
{
    system("cls");
    setlocale(LC_ALL, "Ukrainian");
    int i = 0; // ініціалізуємо лічильник циклу.
    int sum = 0; // ініціалізуємо лічильник суми.
    while (i < 1000)
    {
        i++;
        sum += i;
    }
    cout << "Сума чисел от 1 до 1000 = " << sum << endl;
    return 0;
}
```

У даній програмі в якості умови задається строга нерівність $i < 1000$, лічильник ініціалізується нулем. Так як в циклі while відбувається на одну ітерацію більше, тому він буде виконуватися, до тих пір, поки значення лічильника перестане задовольняти умові, але дана ітерація все одно виконається. Як би ми поставили нестрогу нерівність, то цикл би закінчився, коли змінна i стала б дорівнювати 1001, і таким чином програма виконалася б на 1 ітерацію більше, ніж потрібно за умовою.

Спочатку ми ініціалізували лічильник циклу і змінну, що зберігає суму чисел. В даному випадку ми обов'язково повинні присвоїти лічильнику циклу якесь значення. Далі ми описуємо умову циклу – Поки змінна i менше – виконуй цикл.

При кожній ітерації циклу значення змінної-лічильника i збільшується на одиницю всередині циклу. Коли виконається 1000 ітерацій циклу, лічильник стане рівним 999, і наступна ітерація вже не виконається, оскільки 1000 не менше 1000. Вираз

$sum += i$ ідентично виразу $sum = sum + i$.

Після закінчення виконання циклу виводимо повідомлення з відповіддю.

Оператор циклу є цикл з післяумовою **do while**.

синтаксис:

do { оператори } **while** (умова);

Відмінність від попередньої форми циклу **while** полягає у тому, що умова перевіряється після виконання тіла циклу.

Приклад 2. Обчислити суму цілих чисел від 1 до 1000.

```
int main()
{
    system("cls");
    setlocale(LC_ALL, "Ukrainian");
    int i = 0; // ініціалізуємо лічильник циклу.
    int sum = 0; // ініціалізуємо лічильник суми
    do // виконуємо цикл
    {
        i++;
        sum += i;
    }
    while (i < 1000); // поки виконується умова.
    cout << "Сума чисел від 1 до 1000 = " << sum << endl;
    return 0;
}
```

Питання для актуалізації опорних знань

1. В чому полягає призначення операторів циклів **while** та **do while** ?
2. Які параметри циклів **while** та **do while** мають бути задані?
3. Які форми оператора **for** ви знаєте?
4. В чому відмінність між операторами циклів **while** та **do while**?
5. У чому особливість **while** та **do while**?
6. Який алгоритм виконання таких циклів?