

Кожевніков Ілля КН-923в
Лабораторна робота №5.
Циклічні конструкції.

Завдання 23. (**) Визначити, чи є задане ціле число простим.

Результат

```
kozhevka@AbobaLaptop:~/programming-kozhevnikov/lab05/build$ ./main.bin
Введіть ціле число: 6
6 не є простим числом. Будь ласка, спробуйте ще раз.
Введіть ціле число: 5
5 є простим числом.
kozhevka@AbobaLaptop:~/programming-kozhevnikov/lab05/build$ ./main.bin
Введіть ціле число: 10
10 не є простим числом. Будь ласка, спробуйте ще раз.
Введіть ціле число: а
Помилка введення. Будь ласка, введіть ціле число.
Введіть ціле число: va
Помилка введення. Будь ласка, введіть ціле число.
Введіть ціле число: 11
11 є простим числом.
kozhevka@AbobaLaptop:~/programming-kozhevnikov/lab05/build$ ./main.bin
Введіть ціле число: 111
111 не є простим числом. Будь ласка, спробуйте ще раз.
Введіть ціле число: 123
123 не є простим числом. Будь ласка, спробуйте ще раз.
Введіть ціле число: 80
80 не є простим числом. Будь ласка, спробуйте ще раз.
Введіть ціле число: 74
74 не є простим числом. Будь ласка, спробуйте ще раз.
Введіть ціле число: 7
7 є простим числом.
kozhevka@AbobaLaptop:~/programming-kozhevnikov/lab05/build$ |
```

1. Написав функцію яка повертає бінарне(0/1)=(false/true) значення чи являється число простим:

```
// Функція для перевірки, чи є число простим
int isPrime(int num)
{
    if (num <= 1)
    {
        return 0; // 0 та 1 не є простими числами
    }

    for (int i = 2; i * i <= num; ++i)
    {
        if (num % i == 0)
        {
            return 0; // Знайдено дільник, число не є простим
        }
    }

    return 1; // Якщо жодний дільник не знайдено, число є простим
}
```

2. Написав цикл do-while який валідує інпут користувача. Та якщо введене значення є числом перевіряє його вже готовим методом isPrime. Якщо isPrime повертає 1(true) то завершує цикл і завершує програму

```
int number;

do {
    // Введення числа в користувача

    printf("Введіть ціле число: ");
    if (scanf("%d", &number) != 1)
    {
        printf("Помилка введення. Будь ласка, введіть ціле число.\n");
        while (getchar() != '\n'); // Очищення буфера введення
    }
    else
    {
        // Перевірка, чи є введене число простим
        if (isPrime( num: number))
        {
            printf("%d є простим числом.\n", number);
            break; // Вихід з циклу, оскільки було введено підходяще число
        } else
        {
            printf("%d не є простим числом. Будь ласка, спробуйте ще раз.\n", number);
        }
    }
} while (1); // Безкінечний цикл, який виходить лише при введенні простого числа
```

Відповіді на питання:

1. Як записується і як працює оператор for?

Оператор for використовується для створення циклу, в якому ініціалізується змінна, встановлюється умова та ітерація.

2. Відмінність оператора while від оператора do ... while:

Оператор while виконує тіло циклу, якщо умова істинна перед кожною ітерацією.

3. Як програмуються циклічні алгоритми з явно заданою кількістю повторень циклу?

Оператор do ... while виконує тіло циклу хоча б один раз, навіть якщо умова не є істинною.

4. Як програмуються циклічні алгоритми із заздалегідь невідомим числом повторень циклу?

Циклічні алгоритми з явно заданою кількістю повторень: Використовують оператор for або while з фіксованою умовою завершення.

5. **Напишіть оператор циклу, який не виконується жодного разу.**

```
Int impossible = 1
While (impossible == 0)
{
    ...невиконуємий код
}
```

6. **Напишіть оператор циклу, який виконується необмежену кількість раз:**

Оператор циклу, який не виконується ні разу: Може бути представлений, наприклад, за допомогою for (int i = 0; i < 0; ++i).

7. **Замініть фрагмент програми з оператором for рівнозначним фрагментом програми з оператором while.**

Перед while ініціалізувати int counter = 0. У циклі виконувати зоодно counter++. І умовою цикла зробити while(counter <= cycleCount).

8. **Замініть фрагмент програми з оператором for рівнозначним фрагментом програми з оператором do while.**

Заміна оператора for на оператор do while: Замінюється на еквівалентний оператор do ... while, де ініціалізація робиться перед циклом, тіло циклу виконується хоча б один раз, а умова перевіряється після кожної ітерації.

9. **Як можна перервати виконання оператора циклу?**

Використовується оператор break, який припиняє виконання циклу та переходить до наступного оператора після циклу.

10. **Призначення операторів break і continue:**

break: Використовується для негайного виходу з циклу.

continue: Пропускає поточну ітерацію циклу і переходить до наступної ітерації.