

ЛАБОРАТОРНА РОБОТА № 10

Вкладені цикли

Мета: набуття навичок роботи з різними типами операторів циклу.

Хід роботи:

Завдання 1. Написати програму:

13	Обрахувати $\sum_{i=1}^{50} \sum_{j=1}^{10} 1/\cos(i + j)$ за допомогою вкладених циклів
----	--

Рис. 1. Завдання для написання першої програми

Лістинг програми:

```
#include <stdio.h>
#include <math.h>
#include <windows.h>

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    double r = 0;
    for (int i = 1; i <= 50; i++) {
        for (int j = 1; j <= 10; j++)
            r = r + (1 / cos(i + j));
    }
    printf("Результат = %f\n", r);
    return 0;
}
```

Результат виконання програми:

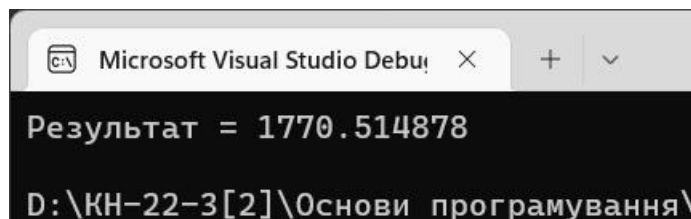


Рис. 2. Результат виконання першої програми

					ДУ «Житомирська політехніка».22.122.13.000 – Лр10			
Змн.	Арк.	№ докум.	Підпис	Дата	Звіт з лабораторної роботи			
Розроб.	Черниш М.							
Перевір.	Терещук С.О.							
Керівник								
Н. контр.								
Зав. каф.					ФІКТ Гр. КН-22-3[2]			
					Лім.	Арк.	Аркушів	
						1	15	

Завдання 2. Написати програму:

13	Довгожитель (вік не менше 100 років) виявив одного разу, що якщо до суми квадратів цифр його віку додати число дня його народження, то як раз вийде його вік. Скільки років довгожителю і якого числа він народився?
----	--

Рис. 3. Завдання для написання другої програми

Лістинг програми:

```
#include <stdio.h>
#include <math.h>
#include <windows.h>

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    int y, n;
    for (int i = 100; i <= 999; i++) {
        for (int j = 1; j <= 31; j++) {
            y = (pow(i / 100, 2) + pow((i / 10) % 10, 2) + pow(i % 10, 2)) + j;
            n = j;
            if (y == i) printf("Вік довгожителя: %d\nЧисло народження довгожителя: %d\n", y, n);
        }
    }
    return 0;
}
```

Результат виконання програми:

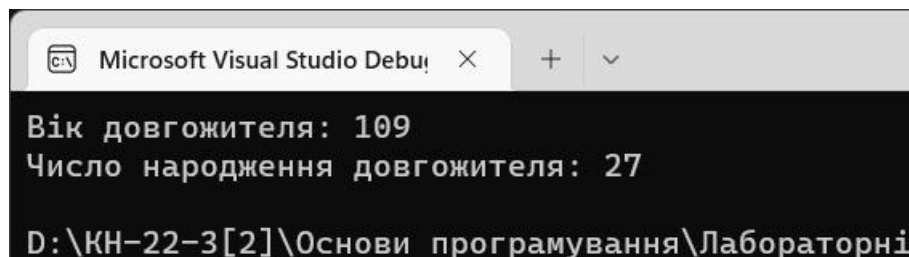


Рис. 4. Результат виконання другої програми

		Черниш М.			ДУ «Житомирська політехніка».22.122.13.000 – Лр10	Арк.
		Терещук С.О.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання на самостійну роботу:

Завдання 1. Поясніть наступні фрагменти коду:

1	<pre>int num; while(scanf("%d",&num) == 1 && num != 0) {printf("%d\n",num); }</pre>	3	<pre>int num; while(scanf("%d",&num) == 1) {if(num == 0) break; printf("%d ",num); }</pre>
2	<pre>int exit = 0, m; for(int n = 0;n < 100 && !exit; n += 1) {scanf("%d",&m); if(m == 0) exit = 1; printf("n*m = %d\n",n*m); }</pre>	4	<pre>int in=0; for(;); { in++; if(in > 100) break; printf("%d ",in); }</pre>

Рис. 5. Фрагменти коду першого завдання самостійної роботи

1) Лістинг програми:

```
#include <stdio.h>
#include <math.h>

int main()
{
    int num;
    while (scanf_s("%d", &num) == 1 && num != 0)
    {
        printf("%d\n", num);
    }
    return 0;
}
```

Результат виконання програми:

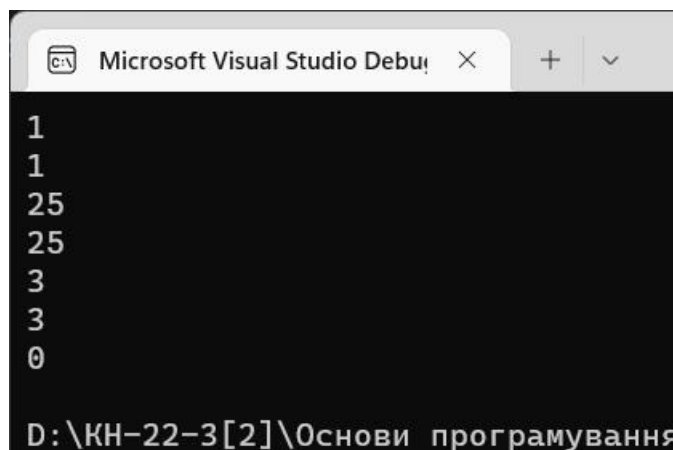


Рис. 6. Результат виконання першого фрагмента коду

Пояснення: 1) В циклі while перевіряється дві умови: поки користувач вводить число відмінне від нуля, то програма буде виводити на екран це число. Якщо умова хибна, тобто користувач вводить нуль, то здійснюється вихід з циклу.

2) Лістинг програми:

```
#include <stdio.h>
#include <math.h>

int main()
{
    int exit = 0, m;
    for (int n = 0; n < 100 && !exit; n += 1)
    {
        scanf_s("%d", &m);
        if (m == 0) exit = 1;
        printf("n*m = %d\n", n * m);
    }
    return 0;
}
```

Результат виконання програми:

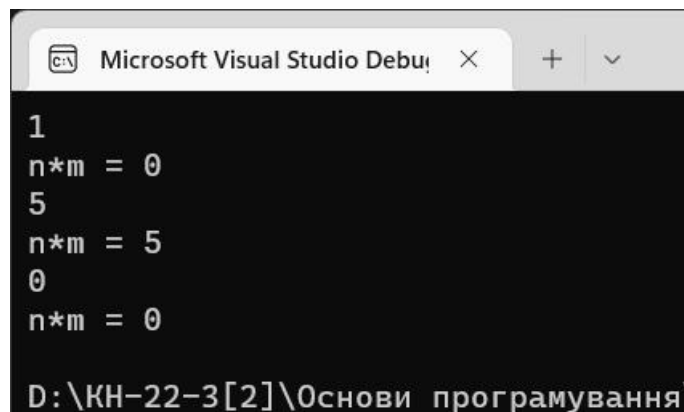


Рис. 7. Результат виконання другого фрагмента коду

Пояснення: 2) Спочатку змінна 'exit' ініціалізується нулем. Потім відбувається вхід в цикл for. В циклі змінна 'n' ініціалізується нулем, перевіряється умова: змінна 'n' менше ста і не дорівнює нулю, тобто істинна. Користувач з клавіатури вводить число, якщо воно дорівнює нулю, то змінна 'exit' присвоює одиницю. Після цього на екран виводиться результат добутку змінних: $n * m$. Потім відбувається зміна лічильника кроком: $n + 1$. І знову повторюється цикл, поки користувач не введе нуль. Інакше цикл буде повторюватися 100 разів.

3) Лістинг програми:

```
#include <stdio.h>
#include <math.h>

int main()
{
    int num;
    while (scanf_s("%d", &num) == 1)
    {
```

		Черниш М.			ДУ «Житомирська політехніка».22.122.13.000 – Лр10	Арк.
		Терещук С.О.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        if (num == 0) break;
        printf("%d ", num);
    }
    return 0;
}

```

Результат виконання програми:

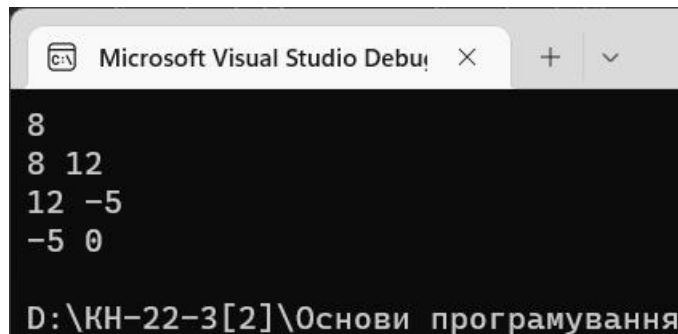


Рис. 8. Результат виконання третього фрагмента коду

Пояснення: 3) Спочатку ініціалізується змінна 'num'. Потім в циклі while перевіряється умова: поки користувач з клавіатури вводить число, яке дорівнює одиниці, тобто істинна, то цикл буде працювати. Якщо введене число – нуль, то виконується оператор break, який перериває цикл. Інакше на екран виводиться це число.

4) Лістинг програми:

```

#include <stdio.h>
#include <math.h>

int main()
{
    int in = 0;
    for (; ; )
    {
        in++;
        if (in > 100) break;
        printf("%d ", in);
    }
    return 0;
}

```

Результат виконання програми:

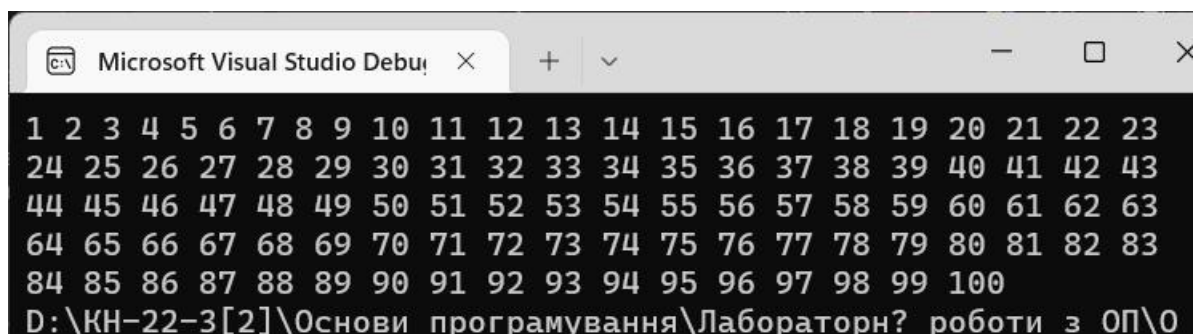


Рис. 9. Результат виконання четвертого фрагмента коду

Пояснення: 4) Спочатку змінна 'in' ініціалізується нулем. Потім відбувається вхід в цикл for. Оскільки в циклі відсутні всі три компоненти, то він є нескінченним. Змінна 'in' збільшується на одиницю через операцію інкременту. Якщо змінна 'in' більше ста, то виконується оператор break, який перериває цикл. Інакше програма виведе на екран числа від одиниці до ста.

Завдання 2. Напишіть програми та побудуйте блок-схеми до кожного завдання:

1	Порахувати, скільки разів зустрічається певна цифра у введеній послідовності чисел. Кількість чисел, що вводяться і цифра, яку необхідно порахувати, задаються введенням з клавіатури.
---	--

Рис. 10. Завдання для написання другої програми самостійної роботи

Лістинг програми:

```
#include <stdio.h>
#include <math.h>
#include <windows.h>

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    int c, n, x, count = 0;
    printf("Введіть кількість чисел: ");
    scanf_s("%d", &c);
    printf("Введіть цифру від 0 до 9, яку необхідно порахувати: ");
    scanf_s("%d", &n);
    printf("Введіть послідовність чисел: \n");
    for (int i = 0; i < c; i++) {
        scanf_s("%d", &x);
        if (x < 0) x = -x;
        while (x != 0) {
            if (x % 10 == n) count++;
            x = x / 10;
        }
    }
    printf("Кількість повторень цифри %d = %d\n", n, count);
    return 0;
}
```

Результат виконання програми:

```

Microsoft Visual Studio Debug Console
Введіть кількість чисел: 5
Введіть цифру від 0 до 9, яку необхідно порахувати: 7
Введіть послідовність чисел:
237
-807
24
277
-7
Кількість повторень цифри 7 = 5
D:\КН-22-3[2]\Основи програмування\Лабораторні роботи з ОП

```

Рис. 11. Результат виконання другої програми самостійної роботи

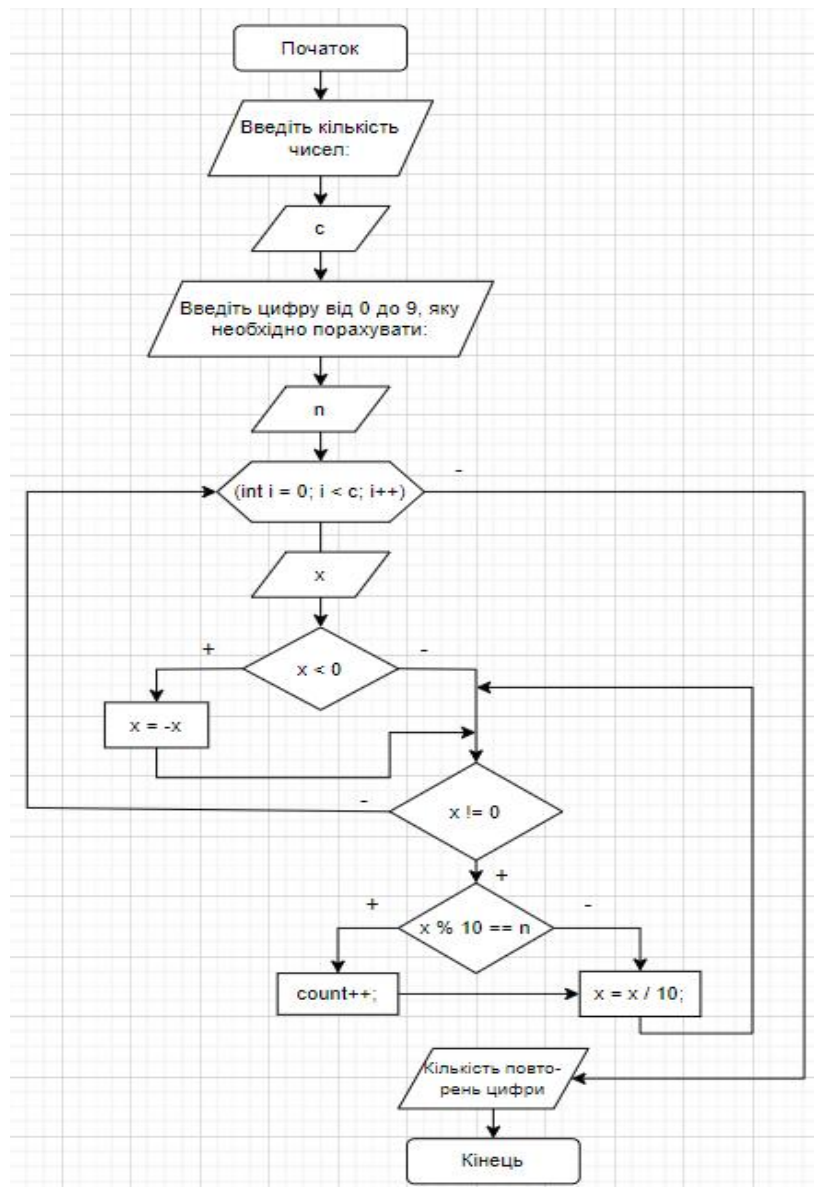


Рис. 12. Блок-схема до другої програми самостійної роботи

2	<p>Довести гіпотезу Сіракуз на діапазоні чисел. Гіпотеза Сіракуз стверджує, що будь-яке натуральне число зводиться до одиниці в результаті повторення таких дій над самим числом і результатами цих дій.</p> <ul style="list-style-type: none"> – Якщо число парне слід розділити його на 2. – Якщо непарне, то помножити його на 3, додати 1 і розділити на 2.
---	---

Рис. 13. Завдання для написання третьої програми самостійної роботи

Лістинг програми:

```
#include <stdio.h>
#include <math.h>
#include <windows.h>

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    int n, c, x;
    printf("Введіть перше натуральне число діапазону: ");
    scanf_s("%d", &n);
    printf("Введіть останнє натуральне число діапазону: ");
    scanf_s("%d", &c);
    for (int i = n; i <= c; i++) {
        x = i;
        printf("-> %d\n", i);
        while (x != 1) {
            if (x % 2 == 0) x = x / 2;
            else x = (x * 3 + 1) / 2;
            printf("%5d", x);
        }
        printf("\n");
    }
    return 0;
}
```

Результат виконання програми:

```
Microsoft Visual Studio Debug Console
Введіть перше натуральне число діапазону: 50
Введіть останнє натуральне число діапазону: 53
-> 50  25  38  19  29  44  22  11  17  26  13  20  10  5  8  4  2  1
-> 51  77 116  58  29  44  22  11  17  26  13  20  10  5  8  4  2  1
-> 52  26  13  20  10  5  8  4  2  1
-> 53  80  40  20  10  5  8  4  2  1
D:\КН-22-3[2]\Основи програмування\Лабораторні роботи з ОП\OP_Lab_10\x64\Debug\Samostiina
```

Рис. 14. Результат виконання третьої програми самостійної роботи

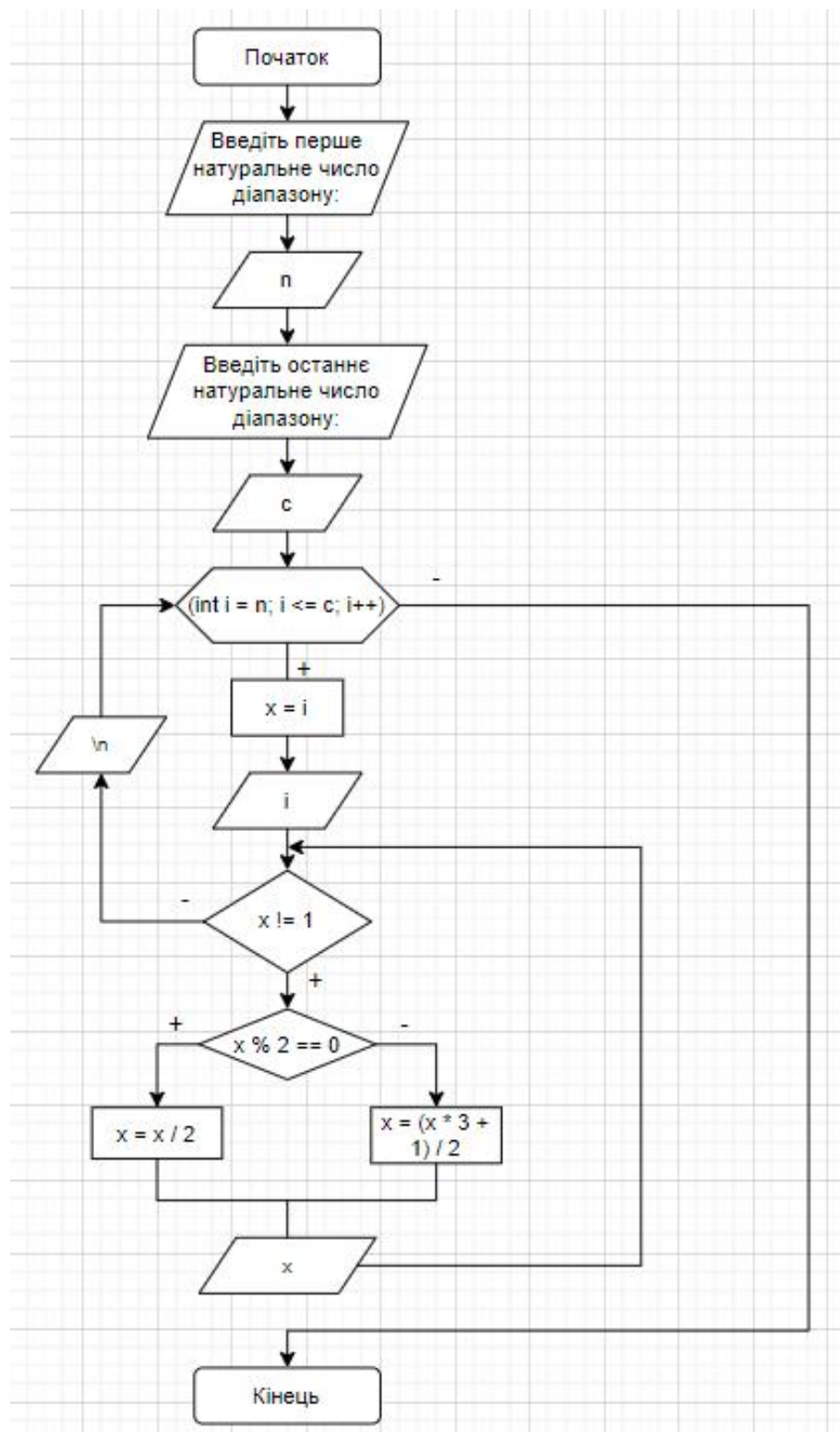


Рис. 15. Блок-схема до третьої програми самостійної роботи

3

Серед натуральних чисел, які були введені, знайти найбільше за сумою цифр. Вивести на екран це число і його суму.

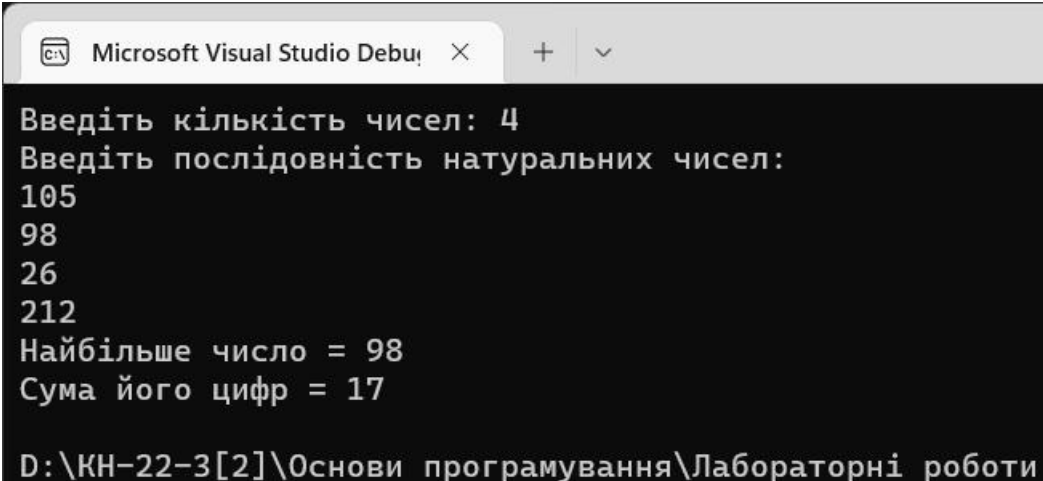
Рис. 16. Завдання для написання четвертої програми самостійної роботи

Лістинг програми:

```
#include <stdio.h>
#include <math.h>
#include <windows.h>

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    int c, n, max = 0, m = 0, j;
    printf("Введіть кількість чисел: ");
    scanf_s("%d", &c);
    printf("Введіть послідовність натуральних чисел: \n");
    for (int i = 0; i < c; i++) {
        int summ = 0;
        scanf_s("%d", &n);
        j = n;
        while (n > 0) {
            summ += n % 10;
            n = n / 10;
        }
        if (summ > max) {
            max = summ;
            m = j;
        }
    }
    printf("Найбільше число = %d\nСума його цифр = %d\n", m, max);
    return 0;
}
```

Результат виконання програми:



```
Microsoft Visual Studio Debug Console
Введіть кількість чисел: 4
Введіть послідовність натуральних чисел:
105
98
26
212
Найбільше число = 98
Сума його цифр = 17
D:\КН-22-3[2]\Основи програмування\Лабораторні роботи
```

Рис. 17. Результат виконання четвертої програми самостійної роботи

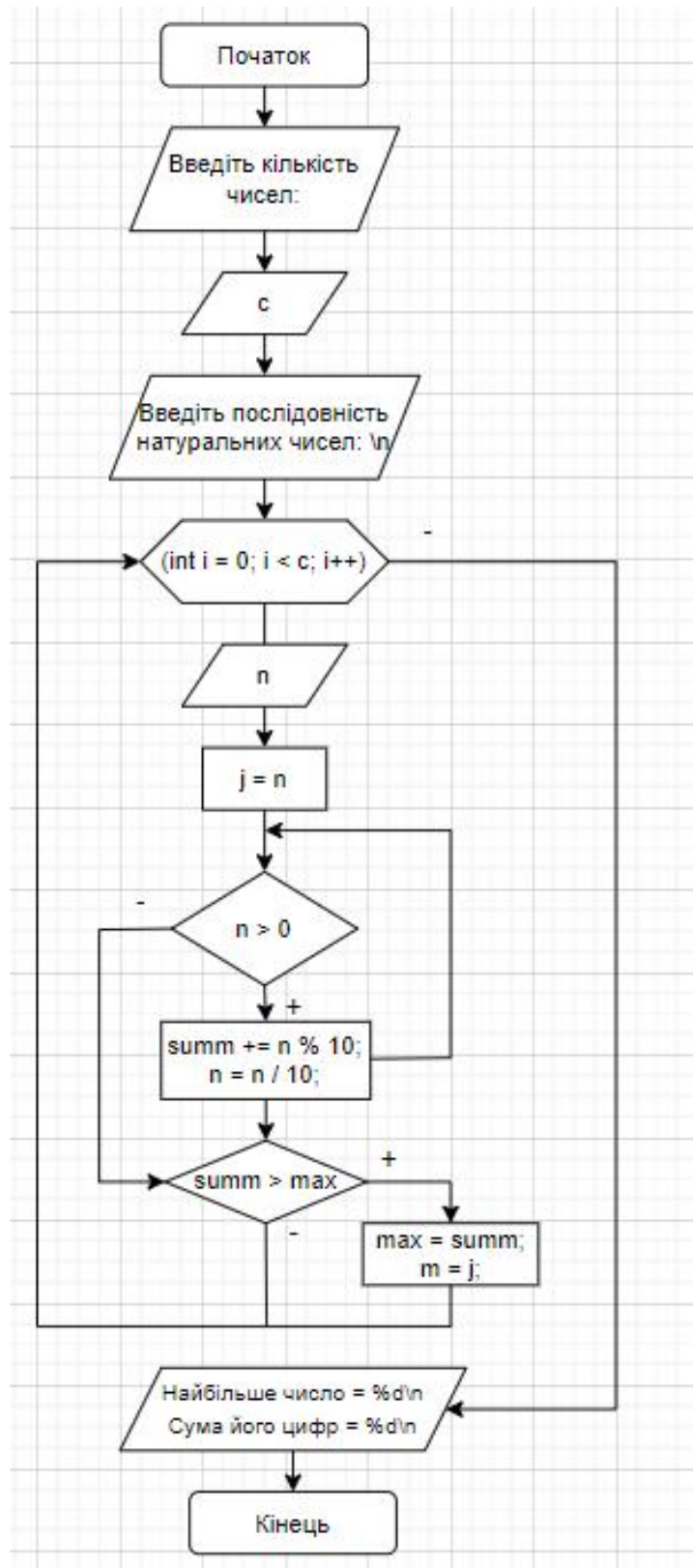


Рис. 18. Блок-схема до четвертої програми самостійної роботи

4	Вивести який-небудь символ по діагоналі уявного квадрата
---	--

Рис. 19. Завдання для написання п'ятої програми самостійної роботи

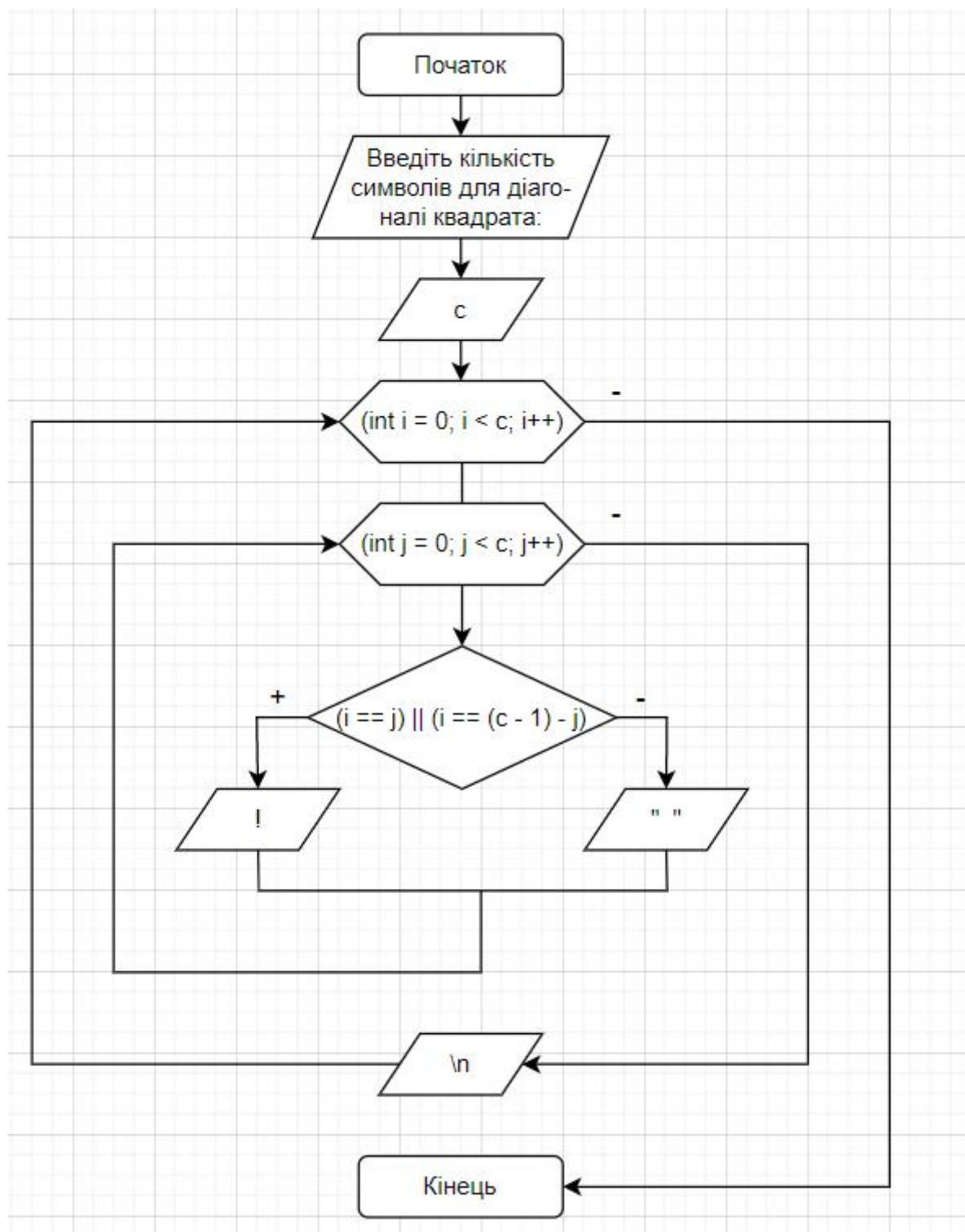


Рис. 20. Блок-схема до п'ятої програми самостійної роботи

Лістинг програми:

```
#include <stdio.h>
#include <math.h>
#include <windows.h>

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    int c;
    printf("Введіть кількість символів для діагоналі квадрата: ");
    scanf_s("%d", &c);
    for (int i = 0; i < c; i++) {
        for (int j = 0; j < c; j++) {
            if (i == j || i == (c - 1) - j) printf("!");
            else printf(" ");
        }
        printf("\n");
    }
    return 0;
}
```

Результат виконання програми:



Рис. 21. Результат виконання п'ятої програми самостійної роботи

		Черниш М.			ДУ «Житомирська політехніка».22.122.13.000 – Лр10	Арк.
		Терещук С.О.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

5

Вивести на екран, прості множники з яких складається введене натуральне число n.

Рис. 22. Завдання для написання шостої програми самостійної роботи

Лістинг програми:

```
#include <stdio.h>
#include <math.h>
#include <windows.h>

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    int n, s, i;
    printf("Введіть натуральне число: ");
    scanf_s("%d", &n);
    printf("-> %d\n", n);
    for (int j = 2; j <= n; j++) {
        if (n % j == 0) {
            s = n / j;
            n = s;
            printf("%-4d", j);
        }
        else j++;
    }
    printf("\n");
    return 0;
}
```

Результат виконання програми:

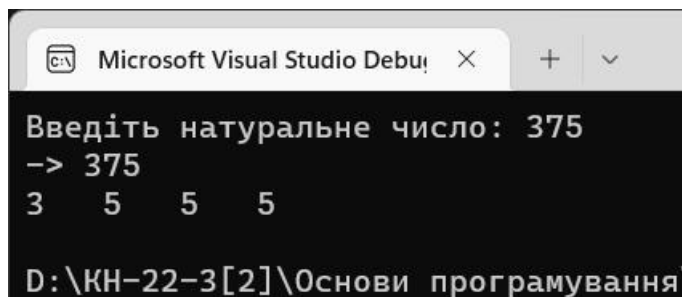


Рис. 23. Результат виконання шостої програми самостійної роботи

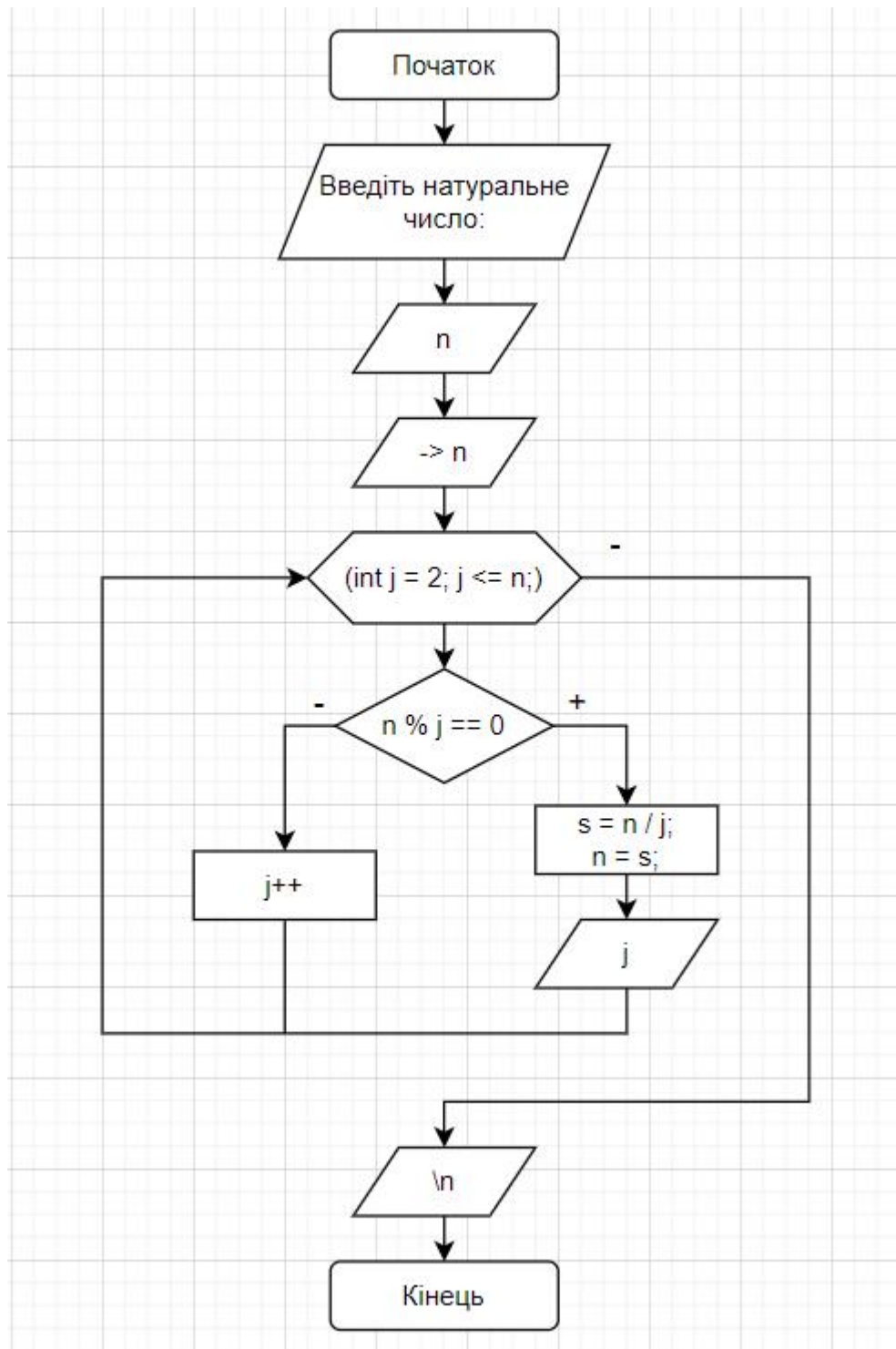


Рис. 24. Блок-схема до шостої програми самостійної роботи