

Лабораторна робота 2_1

Освоєння технології розв'язку задач з використанням циклів на C/C++

Мета роботи

Ознайомитись з операторами циклів мови C++ та отримати практичні навички застосування циклічних програмних кодів.

Хід роботи

- 1) Ознайомитись з методичними вказівками до лабораторної роботи та темою "Цикли";
- 2) відповідно до завдання розробити схему алгоритму програми, яка за допомогою різних операторів циклу виконує: розрахунок суми елементів послідовності, що задано формулою; пошук всіх простих чисел у вказаному діапазоні; пошук корнів рівняння;
- 3) за схемою алгоритму написати програму на мові C++;
- 4) зробити висновки;
- 5) підготувати звіт про виконання лабораторної роботи, що включає наступні пункти: номер, тема, мета та хід лабораторної роботи, завдання, формалізацію змісту задачі, схема алгоритму програми, лістинг програми з коментуванням кожної інструкції, результат роботи програми (скріншот), висновки.

Теоретичні відомості та рекомендації до виконання

Цикли існують як у житті, так і в програмах. До них відносяться дії або події, які повторюються з певною регулярністю. Оператори циклів мови C++ організовують такі дії, зображуючи їх у вигляді правила затримки та повторюваного коду. Циклічні коди дозволяють виконувати математичні розрахунки, пов'язані з послідовністю значень (середнє арифметичне, сума, добуток, пошук факторіала тощо), працювати з масивами (пошук мінімального і максимального значень, упорядкування тощо), використовувати програму протягом будь-якого проміжку часу (заиклення тіла головної функції) і багато інших операцій.

При виконанні завдання до лабораторної роботи студент повинен ознайомитись з операторами циклів і навчитися їх використовувати при створенні проектів на мові C++ (Лекція 9, МВ до ЛР 1_6).

Виконання лабораторної роботи починається ознайомленням з теоретичними відомостями про оператори циклів з перед-та постумовою. Потім студент приступає до розробки схеми алгоритму програми, що виконує три різні операції, відповідно до завдання за варіантом. За схемою алгоритму виконується написання програми мовою C++.

У висновках до звіту студент повинен розкрити призначення теми "Циклічні дії у програмах на основі операторів циклу". Висновок має бути поданий в формі пояснення сфери застосування вивченого матеріалу згідно із розумінням його студентом.

Завдання для самостійного виконання

Розробити схему алгоритму та написати програму, що на основі трьох різних операторів циклу виконує (табл. 3.1):

1) формування послідовності, що задано правилом та діапазоном за варіантом, та розрахунок суми її елементів. Результат виконання вивести на консоль у формі:

$$A_1 + A_2 + \dots + A_{max-1} + A_{max} = \sum_{n=1}^{\max} A_n$$

2) пошук всіх простих чисел у вказаному за варіантом діапазоні і вивід отриманої послідовності на консоль;

3) пошук та вивід на консоль перших ненульових коренів а, b та с рівняння, що задано за варіантом.

Варіанти завдань

Таблиця 3.1

| № вар. | Правило формування послідовності | Діапазон значень (m) | Рівняння | Оператор циклу відповідно до номери підзадачі | | |
|-----------|--|----------------------------|----------------------|---|-------|----------|
| | Номер підзадачі | | | for | while | do-while |
| | 1 | 2 | 3 | | | |
| 1 | непарні A_n , кратні 3 | 0..100 | $3a^2-2b+5c=0$ | 1 | 2 | 3 |
| 2 | $A_n=m/2-3$, кратні 5 | 0..200 | $b^2-(a+c)^2=0$ | 2 | 3 | 1 |
| 3 | парні A_n , кратні 13 | 0..300 | $a*b-c^3-9=0$ | 3 | 1 | 2 |
| 4 | $A_n=m+4$, кратні 3 | 0..400 | $(a+5)^2-(2b-c)^2=0$ | 1 | 3 | 2 |
| 5 | непарні A_n , кратні 7 | 0..500 | $a+3c-7b^2=0$ | 3 | 2 | 1 |
| 6 | $A_n=m-3$, кратні 4 | 0..600 | $a^2+3b-4c=0$ | 2 | 1 | 3 |
| 7 | парні A_n , кратні 17 | 0..700 | $(a-b)^2-c^2=0$ | 1 | 2 | 3 |
| 8 | $A_n=2*m-7$, кратні 9 | 100..200 | $a^3-b^2+c=0$ | 2 | 3 | 1 |
| 9 | непарні A_n , кратні 5 | 100..300 | $(a-3)^2-(b+2c)^2=0$ | 3 | 1 | 2 |
| 10 | $A_n=3*m+5$, кратні 4 | 100..400 | $2a+5c^2-11b=0$ | 1 | 3 | 2 |
| 11 | парні A_n , кратні 15 | 100..500 | $2a^2-7b-3c=0$ | 3 | 2 | 1 |
| 12 | $A_n=2*m+3$, кратні 5 | 100..600 | $(a-c)^2-b^2=0$ | 2 | 1 | 3 |
| 13 | непарні A_n , кратні 11 | 100..700 | $a^3-b*c+11=0$ | 1 | 2 | 3 |
| 14 | $A_n=m+13$, кратні 11 | 100..800 | $(a+b)^2-(c-10)^2=0$ | 2 | 3 | 1 |
| 15 | парні A_n , кратні 7 | 200..300 | $7a^2-9c+7b^2=0$ | 3 | 1 | 2 |
| 16 | $A_n=m+7$, кратні 3 | 200..400 | $a^2-5b+8c=0$ | 1 | 3 | 2 |
| 17 | непарні A_n , кратні 17 | 200..500 | $(b+c)^2-a^2=0$ | 3 | 2 | 1 |
| 18 | $A_n=m/3-5$, кратні 11 | 200..600 | $a^2-3b^2-2c=0$ | 2 | 1 | 3 |
| 19 | парні A_n , кратні 5 | 200..700 | $(3a+2)^2-(b-c)^2=0$ | 1 | 2 | 3 |
| 20 | $A_n=m-11$, кратні 7 | 200..800 | $4a+c^3-3b^2=0$ | 2 | 3 | 1 |
| 21 | непарні A_n , кратні 13 | 200..900 | $4a^2+7b-11c=0$ | 3 | 1 | 2 |
| 22 | $A_n=m+15$, кратні 3 | 300..400 | $(a+b)^2-c^2=0$ | 1 | 3 | 2 |
| 23 | парні A_n , кратні 11 | 300..500 | $a^3-(b+3)^2-c=0$ | 3 | 2 | 1 |
| 24 | $A_n=2*m-5$, кратні 11 | 300..600 | $(a-3)^2-(b+2c)^2=0$ | 2 | 1 | 3 |
| 25 | непарні A_n , кратні 19 | 300..700 | $9a-2c-3b^2=0$ | 1 | 2 | 3 |

Приклад виконання завдання

Розробити схему алгоритму та написати програму, що трьома шляхами на основі операторів циклу формує послідовність, задану правилом та діапазоном і розраховує суму її елементів. Результат виконання зобразити за формою:

$$A_1 + A_2 + \dots + A_{\max-1} + A_{\max} = \sum_{i=1}^{\max} A_i$$

| Вар. № | Діапазон значень (m) | Правило формування послідовності |
|--------|----------------------|----------------------------------|
| N | 0..100 | всі непарні |

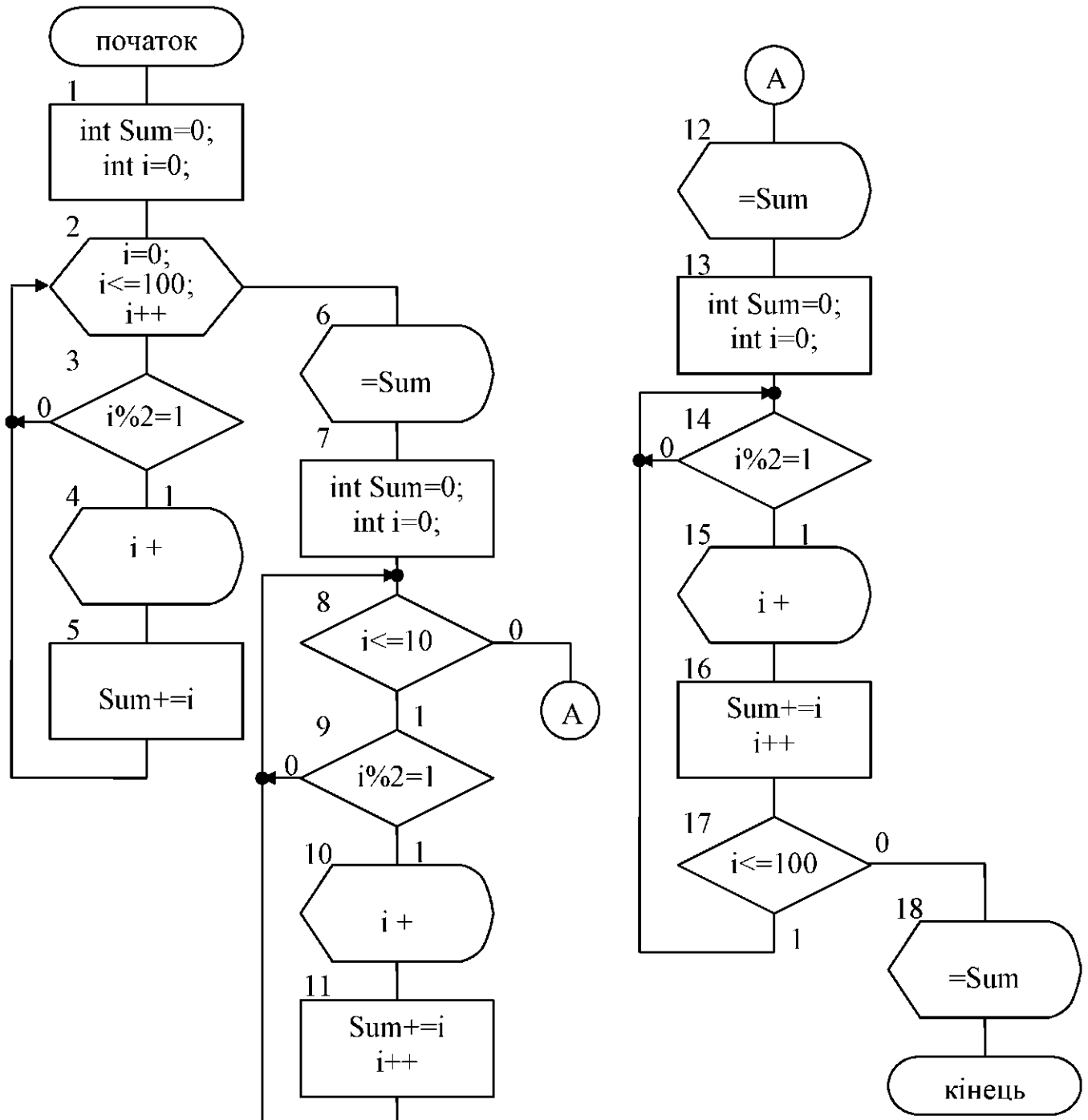


Рис. 3.1. Схема алгоритму програми, що трьома шляхами на основі операторів циклу формує послідовність та рахує суму її елементів

За схемою алгоритму напишемо програмний код

```
#include <stdio.h>                                //підключення бібліотеки вводу/виводу

void main()                                       //підключення бібліотеки математичних функцій
{
    int Sum=0;                                   //оголошення та визначення змінної рахування суми
    int i=0;                                    //оголошення та визначення змінної лічильника
    for(i=0; i<=100; i++)                       //цикл перебору значень від 0 до 100
    {
        if(i%2==1)                             //якщо число непарне
        {
            //якщо поточний елемент останній в послідовності
            if(i+2>100)
                printf("%i", i);                //вивід значення на консоль
            else                                //якщо елемент неостанній в послідовності
                printf("%i+", i);               //вивід значення зі знаком "+"
            Sum+=i;                             //додати значення до суми
        }
    }
    printf("=%i\r\n\r\n", Sum);                //вивід знаку "-" та суми значень

    Sum=0;                                       //перевизначення змінної рахування суми
    i=0;                                        //перевизначення змінної лічильника
    while(i<=100) //цикл, доки значення лічильника менше або дорівнює 100
    {
        if(i%2==1)                             //якщо число непарне
        {
            //якщо поточний елемент останній в послідовності
            if(i+2>100)
                printf("%i", i);                //вивід значення на консоль
            else                                //якщо елемент неостанній в послідовності
                printf("%i+", i);               //вивід значення зі знаком "+"
            Sum+=i;                             //додати значення до суми
        }
        i++;                                    //збільшення значення лічильника на 1
    }
    printf("=%i\r\n\r\n", Sum);                //вивід знаку " " та суми значень

    Sum=0;                                       //перевизначення змінної рахування суми
    i=0;                                        //перевизначення змінної лічильника
    do                                           //початок циклу з постумовою
    {
        if(i%2==1)                             //якщо число непарне
        {
```

//якщо поточний елемент останній в послідовності

```

    if(i+2>100)
        printf("%i", i);           //вивід значення на консоль
    else
        printf("%i+", i);         //якщо елемент неостанній в послідовності
        //вивід значення зі знаком "+"
    Sum+=i;                       //додати значення до суми

    }
    i++;                          //збільшення значення лічильника на 1
} while(i<=100); //цикл доки значення лічильника менше або дорівнює 100
printf( "%i\\r\\n\\r\\n", Sum);   //вивід знаку "=" та суми значень
}

```

Результат виконання програми буде наступним:

1+3+5+7+9+11+13+15+17+19+21+23+25+27+29+31+33+35+37+39+41+43+45+47+49+51+53+55+57+59+61+63+65+67+69+71+73+75+77+79+81+83+85+87+89+91+93+95+97+99=2500

1+3+5+7+9+11+13+15+17+19+21+23+25+27+29+31+33+35+37+39+41+43+45+47+49+51+53+55+57+59+61+63+65+67+69+71+73+75+77+79+81+83+85+87+89+91+93+95+97+99=2500

1+3+5+7+9+11+13+15+17+19+21+23+25+27+29+31+33+35+37+39+41+43+45+47+49+51+53+55+57+59+61+63+65+67+69+71+73+75+77+79+81+83+85+87+89+91+93+95+97+99=2500

Завдання 2

Скласти блок-схему алгоритму та програму, яка обчислює функцію, розкладену в ряд Маклорена з заданою точністю. На друк вивести : функцію, її значення при розкладанні функції в ряд Маклорена, кількість елементів, які врахувались при розрахунках для досягнення заданої точності . Зробити перевірку рішення, використавши стандартну функцію бібліотеки math.h.

Функція та її ряд Маклорена

$$1. (a+x)^m = a^m + m \cdot a^{m-1} \cdot x + \dots + \frac{m(m-1) \dots (m-n+1)}{n!} \cdot a^{m-n} \cdot x^n + \dots, \quad |x| \leq a, \\ m > 0$$

$$2. (a-x)^m = a^m - m \cdot a^{m-1} \cdot x + \dots + (-1)^m \cdot \frac{m(m-1) \dots (m-n+1)}{n!} \cdot a^{m-n} \cdot x^n + \dots, \quad |x| \leq a, \\ m > 0$$

$$3. e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!} + \dots$$

$$4. \cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + (-1)^n \frac{x^{2n}}{(2n)!} + \dots,$$

$$5. \sin x = \frac{x}{1!} - \frac{x^3}{3!} + \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \dots,$$

$$6. \ln x = 2\left(\frac{x-1}{x+1} + \frac{(x+1)^3}{3(x+1)^3} + \dots + \frac{(x-1)^{2n+1}}{(2n+1)(x+1)^{2n+1}} + \dots\right), \quad x > 0$$

$$7. \ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \dots + (-1)^{n+1} \frac{x^n}{n} + \dots, \quad -1 < x \leq 1$$

$$8. \operatorname{arctg} x = x - \frac{x^3}{3} + \frac{x^5}{5} - \dots + (-1)^n \frac{x^{2n+1}}{2n+1} + \dots, \quad |x| < 1$$

$$9. \operatorname{ch} x = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots + \frac{x^{2n}}{(2n)!} + \dots,$$

$$10. a^x = e^{x \ln a} = 1 + \frac{x \ln a}{1!} + \frac{(x \ln a)^2}{2!} + \dots + \frac{(x \ln a)^n}{n!} + \dots$$

| Варіант | Функція | Точність |
|---------|-----------------------------|-----------|
| 1. | $e^{1/2}$ | 10^{-4} |
| 2. | $(2 - 0.34)^5$ | 10^{-5} |
| 3. | $\cos 18$ | 10^{-6} |
| 4. | $\ln(1 + 0.19)$ | 10^{-3} |
| 5. | $(5 + 0.34)^4$ | 10^{-4} |
| 6. | $\ln(1 + 0.04)$ | 10^{-5} |
| 7. | $\operatorname{arctg}(1/5)$ | 10^{-4} |
| 8. | $\ln 2$ | 10^{-7} |
| 9. | $e^{-2/3}$ | 10^{-5} |
| 10. | $e^{-1/2}$ | 10^{-6} |
| 11. | $\operatorname{ch} 0.9$ | 10^{-3} |
| 12. | $\ln(1 + 0.65)$ | 10^{-4} |
| 13. | $\ln 0.32$ | 10^{-6} |
| 14. | $27^{0.45}$ | 10^{-4} |
| 15. | $\operatorname{arctg} 0.65$ | 10^{-5} |
| 16. | $\sin 0.956$ | 10^{-4} |

Питання для підготовки до захисту лабораторної роботи

- 1) Які оператори циклів використовуються для виконання повторюваних операцій в мові C++?
- 2) Чим відрізняються цикли з перед- і постумовами?
- 3) Який формат запису оператора **for**?
- 4) Що таке вкладені цикли?
- 5) Як можна задати вічний цикл?