

## Лабораторна робота № 7

# Вкладені цикли

**Мета роботи:** набути практичних навиків створення програм із вкладеними циклами.

### Теоретичні відомості

Цикли може бути вкладено один в одного. При використанні вкладених циклів треба складати програму в такий спосіб, щоб внутрішній цикл повністю вкладався в тіло зовнішнього циклу, тобто цикли не повинні перетинатися. Своєю чергою, внутрішній цикл може містити власні вкладені цикли. Імена параметрів зовнішнього та внутрішнього циклів мають бути різними. Припускаються такі конструкції:

```
for(k=1; k<=10; k++)
{
    . . .
    for(i=1; i<=10; i++)
    {
        . . .
        for(m=1; m<=10; m++)
        {
            ...
        }
    }
}
```

### Приклади проектів програм із вкладеними циклами

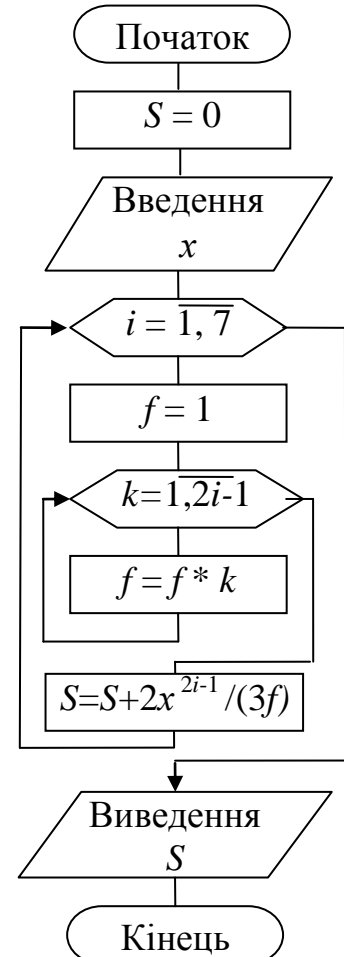
**Приклад 1.** Обчислити суму ряду  $S = \sum_{i=1}^7 \frac{2x^{2i-1}}{3(2i-1)!}$ ,

де  $i = 1, 2, \dots, 7$ .

*Розв'язок.* Для обчислення суми  $S$  треба підсумувати сім доданків, для обчислення кожного з яких слід сформувати вкладений цикл для обчислення факторіалів  $(2i-1)!$ . В наведеній програмі кожний доданок обчислюється в окремій змінній  $u$ .

Текст програми та блок-схема:

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    setlocale(0, ".1251");
    double s=0, u, x;
    int i, k, f;
```



```

cout<<"Введіть значення x = ";
cin>>x;
for (i=1; i<=7; i++)
{ f = 1;
  for (k=1; k<=2*i-1; k++) f *= k;
  u = 2*pow(x,2*i-1)/(3*f);
  s += u;
}
cout<<"\nСума = "<<s;
cin.get();
cin.get();
return 0;
}

```

Результати:

Введіть значення x = 2.45

Сума = 3.83416

**Приклад 2.** Обчислити  $S = \sum_{i=1}^m \frac{i}{i-2} \prod_{k=1}^{i+1} \frac{k+3}{k}$ , значення  $m$  ввести з екрана.

З обчислень вилучити доданки і множники, які дорівнюють нулю в чисельнику або знаменнику.

*Розв'язок.* В цьому прикладі програми наведені цикли є вкладеними один в одного, оскільки параметр внутрішнього циклу  $k$  залежить від параметра зовнішнього циклу  $i$  ( $k$  змінюється від 1 до  $i+1$ ). Добуток  $\prod_{k=1}^{i+1} \frac{k+3}{k}$  є співмножником доданка і обчислюється у внутрішньому циклі у змінній  $P$ . Оскільки внутрішній цикл складається лише з одного оператора, то операторні дужки  $\{ \}$  не є обов'язковими.

Перед зовнішнім циклом для обчислення суми слід обнулити змінну  $S$ , в якій будуть накопичуватись доданки, а перед зовнішнім циклом для обчислення добутку змінній  $P$  слід присвоїти значення 1.

Оскільки при обчислюванні добутку беруть участь лише цілі числа, то, щоб при діленні не втратити дробову частину, слід перетворити чисельник до дійсного типу; для цього можна дописати крапку до числа 3:  $(k+3.0)/k$ .

Перед зовнішнім циклом для обчислення суми слід обнулити змінну  $S$ , в якій будуть накопичуватись доданки, а перед зовнішнім циклом для обчислення добутку змінній  $P$  слід присвоїти значення 1.

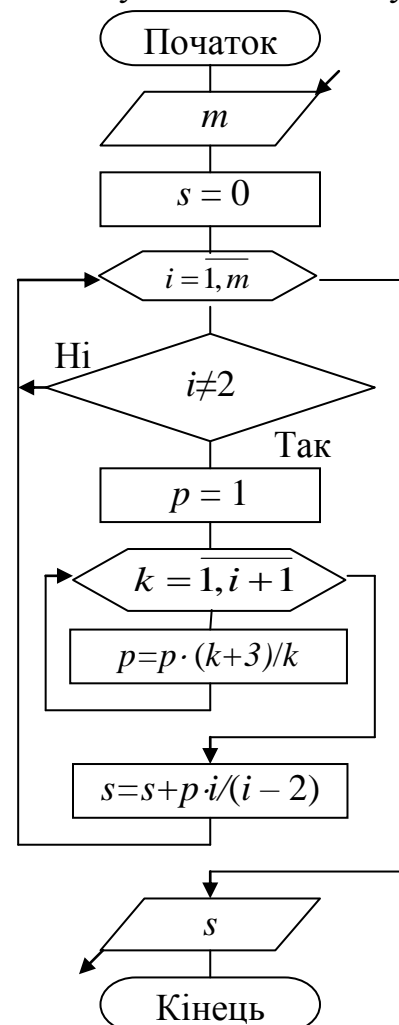
Оскільки при обчислюванні добутку беруть участь лише цілі числа, то, щоб при діленні не втратити дробову частину, слід перетворити чисельник до дійсного типу; для цього можна дописати крапку до числа 3:  $(k+3.0)/k$ .

Текст програми та блок-схема:

```

#include <iostream>
using namespace std;
int main()

```



```
{
    setlocale(0, ".1251");
    int i, k, m;
    cout<<"Введіть значення m = ";
    cin>>m;
    double S = 0, p;
    for (i = 1; i <= m; i++)
        if (i != 2)
        {
            p = 1;
            for(k = 1; k <= i+1; k++)
                p*=(k+3.)/k;
            S += i/(i-2.) * p;
        }
    cout<<"\nСума = "<<S;
    cin.get();
    cin.get();
    return 0;
}
```

Результати:

Введіть значення m = 10

Сума = 1874.05

## Питання та завдання для самоконтролю

- 1) Охарактеризуйте правила організації вкладених циклів.
- 2) Назвіть номер фрагмента програми з вкладеним циклом
  - а) `for(k=1;k<=10;k++)`   б) `for(k=1;k<=10;k++)`   в) `for(k=1;k<=10;k++)`  
`p=k;`   `{ p=k;`   `{ p=k; }`  
`for(j=1;j<=5;j++)`   `for(j=1;j<=5;j++)`   `for(j=1;j<=5;j++)`  
`s+= p*j;`   `s+= p*j; }`   `{ s+= p*j; }`
- 3) Якого значення набуде змінна s після виконання операторів
  - а) `for(s=0,k=1; k<=3; k++)`   б) `for(s=0,k=1; k<=2; k++)`  
`for(j=1; j<=k; j++) s+=j;`   `for(m=k; m<=2; m++) s+=m;`

## Лабораторне завдання

- 1) У протоколі лабораторної роботи дати відповіді на контрольні питання.
- 2) У протоколі лабораторної роботи скласти схеми алгоритмів і написати програми мовою C++ для розв'язання завдань, поданих в табл. 7.1 ... 7.2 відповідно до індивідуального варіанта (див. приклади програм 1 ... 2).
- 3) Створити на комп'ютері програмні проекти у середовищі Visual C++ для реалізації написаних програм. Занести результати обчислень до протоколу.

## Варіанти індивідуальних завдань з теми “Вкладені цикли”

Таблиця 7.1

## Індивідуальні завдання базового рівня складності

(довільне значення  $x$  слід Ввести)

№	Функція	№	Функція	№	Функція	№	Функція
1	$y = \sum_{k=1}^{10} \frac{x^k}{(k+1)!}$	2	$y = \sum_{i=1}^6 \frac{(-1)^i x^{2i}}{(3i-1)!}$	3	$y = \sum_{i=1}^{10} \frac{(-1)^{i+1} i!}{2^{2i-1} \sin x}$	4	$y = \sum_{i=1}^5 \frac{(2i-1)!}{x^{2i-1}}$
5	$y = \sum_{k=1}^7 \frac{k! \cos(\pi k - x)}{\ln x}$	6	$y = \prod_{k=1}^{10} \frac{x^k}{(2k)!}$	7	$y = \sum_{k=1}^7 \frac{(2k-1)!}{2^k x^{k-1}}$	8	$y = \sum_{i=1}^5 (-1)^{i+1} \frac{\cos x^i}{2i!}$
9	$y = \sum_{i=1}^5 (-1)^{i+1} \frac{\sin x^i}{(2i-1)!}$	10	$y = \sum_{i=1}^3 \frac{x^{2i} (2i-1)!}{2^i}$	11	$y = \sum_{i=1}^6 \frac{(-1)^{i+1} x^{2i}}{(2i-1)!}$	12	$y = \sum_{i=1}^9 \frac{(-1)^i \operatorname{tg} x^2}{(2i+1)!}$
13	$y = \sum_{i=1}^{11} \frac{(-1)^i x^i}{i! \cos\left(i + \frac{\pi}{4}\right)}$	14	$y = \sum_{k=1}^6 \frac{k!}{(1+x)^k}$	15	$y = \prod_{k=1}^5 \frac{(k+1)!}{x^{k+2}}$	16	$y = \sum_{k=1}^5 \frac{x^{k+2}}{k!}$
17	$y = \prod_{k=1}^5 \frac{(-1)^k (2k)!}{4.5 x^{2k-1}}$	18	$y = \sum_{i=1}^6 \frac{\operatorname{tg}(x - \pi)^2}{i!}$	19	$y = \sum_{k=1}^8 \frac{(-1)^k x^{2k-1}}{2^k k!}$	20	$y = \sum_{k=1}^5 \frac{(-1)^k x^{3k-2}}{(k+1)!}$
21	$y = \sum_{i=1}^{11} \frac{(-1)^i x^{3i}}{(2i-1)!}$	22	$y = \sum_{k=1}^8 \frac{2k! \cos(\pi - x)}{x^{2k-1}}$	23	$y = \sum_{k=1}^7 \frac{(-1)^k \cdot k!}{\sqrt{x} + \sin x}$	24	$y = \prod_{k=1}^5 \frac{(-1)^k k!}{x^{k+2}}$
25	$y = \sum_{k=1}^6 \frac{(-1)^k \cdot x^{4k+1}}{(2k)!}$	26	$y = \sum_{k=1}^6 (-1)^k \frac{x^k}{k!}$	27	$y = \sum_{k=1}^5 \frac{(-1)^{k-1} \cdot x^{k+2}}{k!}$	28	$y = \sum_{i=1}^{10} \frac{(2i-1)x^{i+1}}{2i!}$
29	$y = \sum_{i=1}^5 \frac{(-1)^i x^{2i}}{(i+1)! \cos x}$	30	$y = \sum_{i=1}^7 \frac{(i+1)x^i}{i!}$				

Таблиця 7.2

## Індивідуальні завдання високого рівня складності

№	Функція	№	Функція	№	Функція
1	$S = \sum_{k=1}^n \frac{(k+1)}{(k-5)} \prod_{m=1}^{k+1} \frac{m-2}{m^2-9}$	2	$Z = \prod_{j=-4}^k \frac{(j+2)}{j-3} \sum_{i=j}^{k+5} \left( \frac{\sqrt[5]{i+5}}{i-11} \right)$	3	$S = \sum_{k=0}^n \frac{(-2)^{k+1}}{(k-5)} \prod_{i=1}^{k+1} \frac{i}{i^2-16}$
4	$Z = \prod_{j=-2}^k \frac{j}{j-1} \sum_{i=j}^k \frac{i}{i+5}$	5	$W = \sum_{i=1}^k \frac{(-1)^i (i+1)!}{i^2-4}$	6	$Y = \sum_{n=1}^k \frac{(-1)^{2-n} (n^2-9)^2}{(n-2)(n+1)!}$
7	$W = \sum_{i=1}^k \frac{(-1)^i}{(i-4)^2} \prod_{n=1}^{i+2} \frac{n^2-4}{n+2}$	8	$L = \prod_{j=1}^k \frac{(j-5)}{j-2} \sum_{i=k}^{12} \frac{\sqrt{i+5}}{i-1}$	9	$Q = \sum_{k=1}^n \frac{(k-1)^{k+1} (k-3)}{(k+1)!}$
10	$Z = \prod_{t=1}^k \frac{k-t-1}{\cos(t)-3} \sum_{i=1}^t \left( \frac{3i-2}{i-7} \right)$	11	$P = \prod_{j=1}^k \frac{(j-6)j}{j-3} \sum_{i=j}^{12} \frac{\sqrt[3]{i+5}}{i-11}$	12	$A = \prod_{j=1}^k \frac{j-3}{(j-4)j} \sum_{i=0}^j \frac{\sqrt{i+4}}{i-1}$
13	$P = \prod_{j=2}^k \frac{(j-6)j}{(j-3)(j-1)!}$	14	$Q = \sum_{k=1}^n \frac{(-1)^k (k-3)^2}{k!}$	15	$U = \prod_{t=1}^k \frac{\sin(t)}{t-3} \sum_{i=1}^t \left( \frac{i+2}{i-7} \right)$

Закінчення табл. 7.2

№	Функція	№	Функція	№	Функція
16	$S = \sum_{k=1}^n \frac{(-3)^{3k-1}}{(k-2)^{3k+1}} \prod_{m=1}^{k+n} \frac{m+3}{m^2-25}$	17	$Z = \prod_{j=-4}^k \frac{j+2}{j(j-3)} \sum_{i=j}^{k+5} \left( \frac{i+5}{i-11} \right)$	18	$Y = \sum_{i=1}^k \frac{(i-4)^i}{(3-i)^2} \prod_{n=i}^{2+k} \frac{n+0.8}{n-i}$
19	$G = \prod_{j=-3}^k \frac{(j-1)^{k+5}}{4j-3} \sum_{i=j}^{k+5} \left( \frac{i+5}{1-i+j} \right)$	20	$D = \sum_{i=-2}^k \frac{(-2^i) \sin^2(i+3)}{(i+3)!}$	21	$R = \sum_{i=0}^k \frac{(1-i)^i}{(i+3)} \prod_{n=i}^{2k} \frac{n-i}{n+2}$
22	$Q = \sum_{k=1}^n \frac{(k-1)^{k+1} (k-7)}{k!}$	23	$W = \sum_{i=1}^k \frac{(-1)^i (i-3)^2}{i!}$	24	$A = \prod_{j=1}^k \frac{(j^2-4)j}{j-k+1} \sum_{i=j}^9 \frac{i-3}{i-7}$
25	$P = \prod_{j=1}^k \frac{j-6}{j-3} \sum_{i=j}^{10} \frac{\sin(i+5)}{i-4}$	26	$Z = \sum_{n=-2}^k \frac{(n+1) n-9 }{(n+3)!}$	27	$p = \sum_{i=1}^n \frac{(2i+1)(i-3)}{(2i-1)!}$
28	$W = \sum_{i=1}^k \frac{(i-1)^i}{(i+3)^2} \prod_{n=1}^{i+1} \frac{n^2-1}{n-2}$	29	$Y = \sum_{i=-1}^k \frac{(k-i)^i (i+2)!}{i^2-4}$	30	$F = \sum_{n=0}^k \frac{(n+2^k) n-4 }{(n)!}$

## Лабораторна робота № 8

# Оператори циклу **while** та **do-while**

**Мета роботи:** набути практичних навиків організації циклічних обчислень у C++ з використанням операторів циклу з передумовою **while** та післяумовою **do-while**.

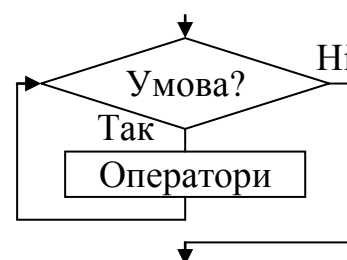
## Теоретичні відомості

Оператори з передумовою та післяумовою використовуються для організації циклів і є альтернативними операторів **for**. Звичайно цикл з передумовою використовується, якщо кількість повторювань заздалегідь є невідома або немає явно вираженого кроку змінювання параметра циклу. А для багаторазових повторювань тіла циклу відомим є вираз умови, за істинності якої цикл продовжує виконання. Цю умову слід перевіряти кожного разу перед черговим повторенням.

Синтаксис циклу з передумовою:

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

*Послідовність операторів (тіло циклу)* виконується, доки *умова* є істинна (**true**, має ненульове значення), а вихід з циклу здійснюється, коли *умова* стане хибною (**false**, матиме нульове значення). Якщо *умова* є хибною при входженні до циклу, то *послідовність операторів* не виконуватиметься жодного разу, а керування передаватиметься до наступного оператора програми.

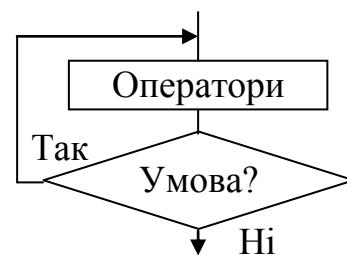


Цикл з післяумовою використовується, якщо є потреба перевіряти умову кожного разу після чергового повторення. Відмінність циклу з передумовою від циклу з післяумовою полягає в першій ітерації: цикл з післяумовою завжди виконується принаймні одноразово незалежно від умови.

Синтаксис циклу з післяумовою:

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

*Послідовність операторів (тіло циклу)* виконується один чи кілька разів, доки *умова* стане хибною (**false** чи дорівнюватиме нулю). Якщо *умова* є істинна (ненульова), то оператори тіла циклу виконуються повторно. Оператор циклу **do-while** використовується в тих випадках, коли є потреба виконати тіло циклу хоча б одноразово, оскільки перевірка умови здійснюється після виконання операторів.



Якщо тіло циклу складається з одного оператора, то операторні дужки **{ }** не є обов'язкові.

Оператори `while` та `do-while` можуть завчасно завершитись при виконанні операторів `break`, `goto`, `return` усередині тіла циклу.

Варто зауважити, що в тілі циклу слід передбачати змінювання параметрів, які беруть участь в умові, інакше умову виходу з циклу ніколи не буде виконано й відбуватиметься зациклювання.

Наведемо відмінності роботи різних операторів циклу на прикладі обчислення суми всіх непарних чисел у діапазоні від 10 до 100:

1) з використанням оператора `for`

```
int i, s=0;
for (i=11; i<100; i += 2) s += i;
```

2) з використанням оператора `while`

```
int s=0, i=11;
while (i<100) { s += i; i += 2; }
```

3) з використанням оператора `do-while`

```
int s=0, i=11;
do { s += i; i += 2;} while (i<100);
```

## Приклади програм

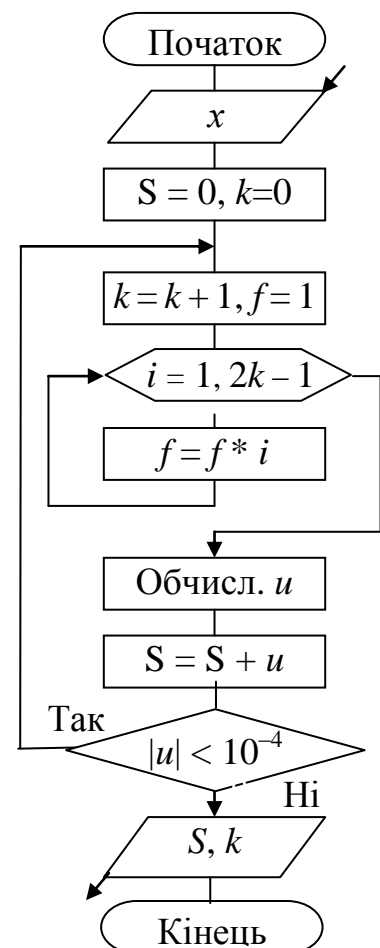
**Приклад 1.** Обчислити суму ряду  $S = \sum_{k=1}^{\infty} \frac{x^{2k+1}}{3^k (2k-1)!}$ , підсумовуючи члени

ряду, значення яких за модулем є більше за задану точність  $\varepsilon = 10^{-4}$ . Визначити кількість доданків. Значення  $x$  ( $-2 < x < 2$ ) вводити з клавіатури.

*Розв'язок.* У цій програмі недоцільно використовувати оператор циклу з параметром `for`, оскільки кількість повторень циклу є наперед невідома. Доцільним буде використання оператора циклу з післяумовою `do-while`, оскільки на момент першої перевірки умови вже треба знати значення першого доданка.

Текст програми та блок-схема:

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    setlocale(0, ".1251");
    double x, f, u, s=0;
    int i, k=0;
    cout << "Введіть значення x= ";
    cin >> x;
    cout << "\nРезультати:\n";
```



```

do                                     // Цикл з післяумовою
{ k++;                                // Збільшення змінної k на 1
  for(i=1,f=1; i<=2*k-1; i++) f *= i; // Обчислення факторіала
  u=pow(x, 2*k+1)/(pow(3.,k)*f);      // Обчислення k-го доданка
  s+=u;                               // Підсумовування доданків
} while(fabs(u)>=1e-4);
cout << "Сума = " << s << endl;
cout << "Кількість доданків = " << k << endl;
system("pause>>void");
return 0;
}

```

Результати роботи:

Введіть значення  $x = 2.5$

Результати:

Сума = 7.21478

Кількість доданків = 6

**Приклад 2.** Обчислити суму ряду  $y = \sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^{2k-1}}{(k+3)(2k)!}$ , підсумовуючи чле-

ни ряду, значення яких за модулем є більше за задану точність  $\varepsilon$ . Визначити кількість доданків. Значення  $x$  ( $-2 < x < 2$ ) і  $\varepsilon = 10^{-4}$  вводити з клавіатури.

*Розв'язок.* Множник  $(-1)^{k+1}$  за непарних  $k = 1, 3, 5, \dots$  дорівнює 1, а за парних  $k = 2, 4, \dots$  – дорівнює  $(-1)$ . Отже, поданий ряд є знакозмінним, де всі парні доданки будуть від'ємними, а всі непарні – зі знаком "+". Для наочності й контролю правильності у розв'язку окремо наведемо всі доданки.

Для того щоби зробити алгоритм програми більш оптимальним, його можна вдосконалити, але для цього слід обчислити рекурентний множник. Це дозволить у даній програмі позбавитись від вкладеного циклу для обчислення факторіала й оператора перевірки на парність  $k$ .

```

u = x/8;  s = u;
do
{
  k++;
  cout<< "доданок " << k << " : " << u << endl;
  r = -x*x*(k+2)/((3+k)*(2*k) *(2*k-1));
  u *= r ;
  s += u;
}
while(fabs(u) >= eps);

```

Зупинимось більш докладно на виведенні рекурентної формули.



Подамо суму ряду  $y = \sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^{2k-1}}{(k+3)(2k)!}$  у вигляді  $y = \sum_{k=1}^{\infty} u_k$ , де

$u_k = \frac{(-1)^{k+1} x^{2k-1}}{(k+3)(2k)!}$ . Рекурентний множник  $R$  – це співвідношення двох поряд розміщених членів ряду:

$$u_2 = R \cdot u_1, \quad u_3 = R \cdot u_2, \quad \dots, \quad u_k = R \cdot u_{k-1},$$

звідки

$$R = \frac{u_k}{u_{k-1}}.$$

Для визначення  $R$  в цю формулу слід підставити  $u_k = \frac{(-1)^{k+1} x^{2k-1}}{(k+3)(2k)!}$  та  $u_{k-1}$ . Для обчислення  $u_{k-1}$  підставимо у вираз для  $u_k$   $(k-1)$  замість  $k$ :

$$\begin{aligned} u_{k-1} &= \frac{(-1)^k x^{2(k-1)-1}}{(k+2)(2k-2)!} = \frac{(-1)^k x^{2k-3}}{(k+2)(2k-2)!}; \\ R = \frac{u_k}{u_{k-1}} &= u_k \cdot \frac{(k+2)(2k-2)!}{(-1)^k x^{2k-3}} = \frac{(-1)^{k+1} \cdot x^{2k-1} \cdot (k+2) \cdot (2k-2)!}{(-1)^k \cdot x^{2k-3} \cdot (k+3) \cdot (2k)!} = \\ &= \frac{(-1)^{k+1-k} \cdot x^{2k-1-(2k-3)} \cdot (k+2)}{(k+3)} \cdot \frac{1 \cdot 2 \cdot \dots \cdot (2k-2)}{1 \cdot 2 \cdot \dots \cdot (2k-2) \cdot (2k-1) \cdot 2k} = -\frac{x^2(k+2)}{2k(k+3)(2k-1)}. \end{aligned}$$

Крім рекурентного множника  $R$ , треба обчислити перший член ряду за  $k=1$ :

$$u_1 = \frac{(-1)^2 x}{4 \cdot (2)!} = \frac{x}{8}.$$

Текст програми:

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    setlocale(0, ".1251");
    double x, u, r, s=0, eps;
    int i, f, k=0;
    cout<<"Введіть значення x= ";
    cin>> x;
    cout<<"Введіть значення точності ";
    cin>> eps;
    cout<< "\nРезультати:\n";
    u = x/8;  s = u;
    do {
        k++;
        cout<< "доданок " << k << " : " << u << endl;
```

```

    r = -x*x*(k+2)/((3+k)*(2*k) *(2*k-1));
    u *= r ;
    s += u;
} while(fabs(u) >= eps);
cout << "Сума = " << s << endl;
cout << "Кількість доданків = " << k << endl;
system("pause>>void");
return 0;
}

```

Результати роботи:

Введіть значення  $x = 2.4$

Введіть значення точності 0.0001

Результати:

доданок 1: 0.3

доданок 2: -0.648

доданок 3: 0.248832

доданок 4: -0.0398131

доданок 5: 0.00351005

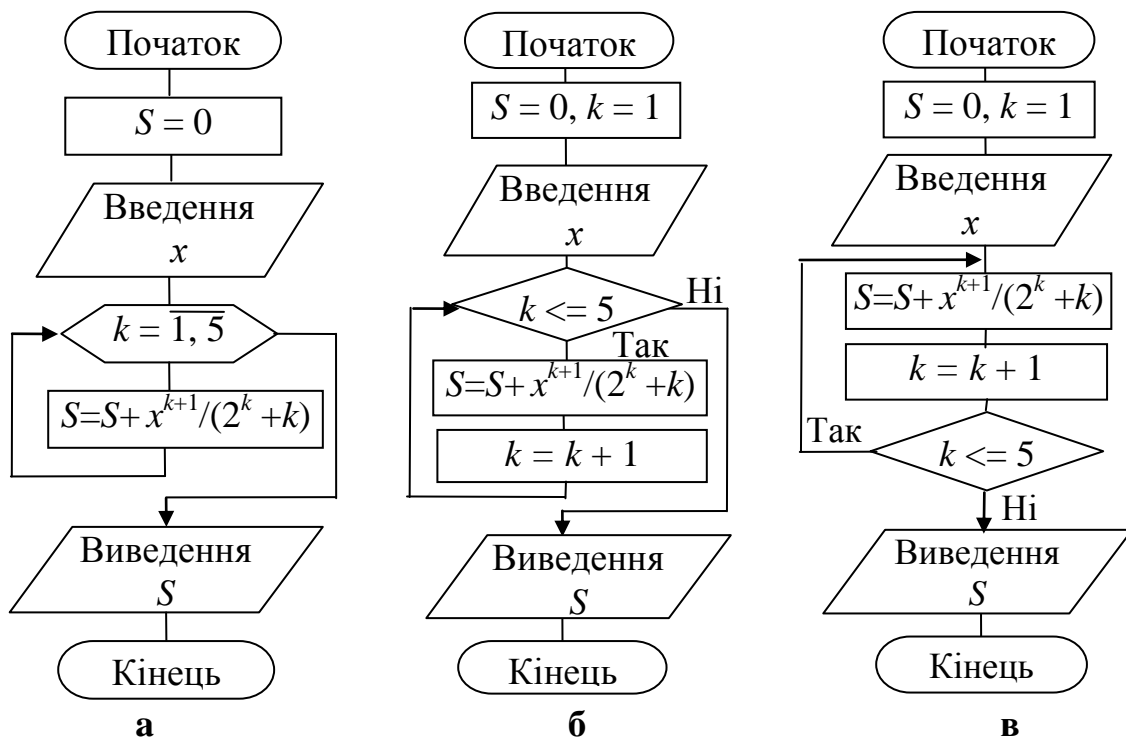
доданок 6: -0.0001196563

Сума = -0.13566

Кількість доданків = 6

**Приклад 3.** Обчислити суму  $f(x) = \sum_{k=1}^7 \frac{x^{k+1}}{2^k + k}$  трьома варіантами, використовуючи різні оператори циклу.

Схеми алгоритмів:



Схеми алгоритмів програм з різними операторами циклу:

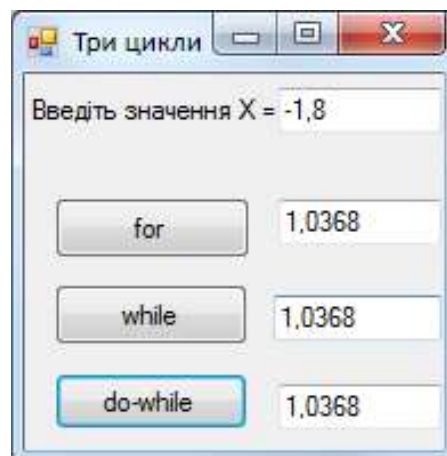
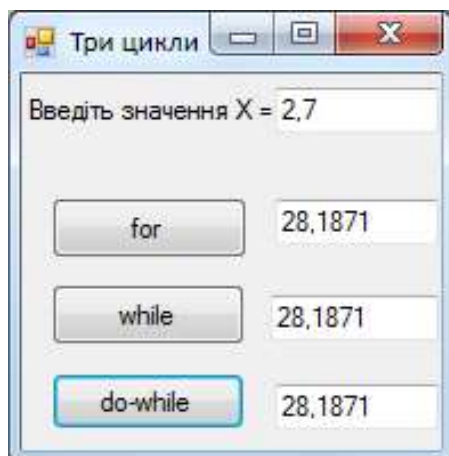
а) **for**; б) **while**; в) **do-while**

Програма з використанням різних операторів циклу:

```
private: System::Void button1_Click(System::Object^ sender,
                                     System::EventArgs^ e)
{
    double S=0, x=Convert::ToDouble(textBox1->Text);
    for(int k=1; k<=5; k++)
        S += Math::Pow(x,k+1)/(Math::Pow(2,k)+k);
    textBox2->Text=S.ToString("0.0000");
}
```

```
private: System::Void button2_Click(System::Object^ sender,
                                     System::EventArgs^ e)
{
    double S=0, x=Convert::ToDouble(textBox1->Text);
    int k=1;
    do {
        S += Math::Pow(x,k+1)/(Math::Pow(2,k)+k);
        k++;
    } while(k<=5);
    textBox3->Text=S.ToString("0.0000");
}
```

```
private: System::Void button3_Click(System::Object^ sender,
                                     System::EventArgs^ e)
{
    double S=0, x=Convert::ToDouble(textBox1->Text);
    int k=1;
    while(k<=5)
    { S += Math::Pow(x,k+1)/(Math::Pow(2,k)+k);
      k++;
    }
    textBox4->Text=S.ToString("0.0000");
}
```



**Приклад 4.** Визначити і вивести всі дільники введеного натурального числа.

Текст програми:

```
#include <iostream>
int main()
{ unsigned int x, i;
  std::cout << "Enter x: ";
  std::cin >> x;
  std::cout << std::endl;
  for (i=1; i<x; i++)
    if (x%i == 0) // Якщо ділиться без остачі,
                 // вивести черговий дільник
      std::cout << i << "\t";
  std::cin.get();
  std::cin.get();
  return 0;
}
```

Enter x: 75

1      3      5      15      25

Enter x: 122

1      2      61

**Приклад 5.** Визначити кількість цифр введеного натурального числа.

Текст програми:

```
#include <iostream>
using namespace std;
int main()
{ unsigned int N, kol;
  cout << "N="; cin >> N;
  for (kol=1; N/10>0; kol++,N/=10);
  cout << "kol=" << kol << endl;
  system("pause>>void");
  return 0;
}
```

N=50026  
kol=5

**Приклад 6.** Визначити найменшу та найбільшу цифру введеного натурального числа.

Текст програми:

```
#include <iostream>
using namespace std;
int main()
{ unsigned int N, z, min, max;
  cout<<"N="; cin>>N;
  for (max=min=N%10; N>0; N/=10)
  { z=N%10;
    if (z>max) max=z;
    if (z<min) min=z;
  }
  cout << "min=" << min << endl << "max=" << max << endl;
  system("pause>>void");
  return 0;
}
```

N=550308  
min=0  
max=8

**Приклад 7.** Перевірити чи є введене натуральне число паліндромом.

Паліндром – це число (слово чи фраза), яке однаково читається в обох напрямках, або, інакше кажучи, будь-який симетричний відносно своєї середини набір символів. Наприклад, числа 404, 12521 – паліндроми.

Текст програми:

```
#include <iostream>
using namespace std;
int main()
{
    setlocale(0, ".1251");
    unsigned long long n,m,d; // int kol, i; bool Fl;
    cout<<"N="; cin>>n;
    d = 0; m = n;
    while ( m!=0)
    { d = d*10 + m%10;
      m /= 10;
    }
    if (d==n) cout<<"Так, число є паліндромом"<<endl;
    else cout<<"Ні, число не є паліндромом "<<endl;
    system("pause>>void");
    return 0;
}
```

```
N=25052
Так, число є паліндромом
```

## Питання для самоконтролю

- 1) Які оператори циклу використовуються у мові C++?
- 2) Назвіть правильну, на Ваш погляд, послідовність номерів, для запису елементів оператора циклу `while`:
  - а) логічний вираз умови;
  - б) оператори тіла циклу;
  - в) `while`.
- 3) Вкажіть значення `n` після виконання фрагментів програми:
  - а) `int k=0, n=17; while(k<7){ k++; n--; }`
  - б) `int m=1, n=1; while(m<5){ m+=2; n=n*m; }`
- 4) Якими є структура і порядок виконання оператора циклу `do-while`?
- 5) Вкажіть значення `y` після виконання фрагментів програми:
  - а) `int i=1, y=1; do {y*=i++;} while(y<7);`
  - б) `int k=1; float y=0; do{k+=2;y+=1./k;}while(k<5);`
- 6) Запишіть трьома операторами циклу варіанти обчислення  $S = \sum_{i=1}^6 i^2$ .
- 7) Вкажіть значення `s` після виконання операторів
 

```
s=0.5; i=0;
while(i<5) i++;
s+=1.0/i;
```

## Лабораторне завдання

1) У протоколі лабораторної роботи дати відповіді на контрольні питання.

2) У протоколі лабораторної роботи скласти схему алгоритму і програму для обчислення суми нескінченного ряду, підсумовуючи члени ряду, значення яких за модулем перевищують задану точність  $\varepsilon = 10^{-4}$ . Визначити кількість доданків. Обчислення виконати для  $x$  ( $-2 < x < 2$ ), яке ввести. Завдання вибрати з табл. 8.1 відповідно до індивідуального варіанта (див. приклад програм 1 ... 2).

3) У протоколі лабораторної роботи скласти схему алгоритму і написати програму мовою C++ для обчислення функції  $f(x)$  трьома варіантами із застосуванням різних операторів циклу (див. приклад 3). Функцію  $f(x)$  вибрати з табл. 8.2.

4) У протоколі лабораторної роботи скласти схему алгоритму і написати програму мовою C++ для розв'язання завдання, наведеного у табл. 8.3 (див. приклади програм 4 ... 7).

5) Створити на комп'ютері програмні проекти у середовищі Visual C++ для реалізації написаних програм. Занести результати обчислень до протоколу.

Таблиця 8.1

### Індивідуальні завдання середнього рівня складності

№	Функція $f(x)$	№	Функція $f(x)$	№	Функція $f(x)$
1	$\sum_{k=1}^{\infty} \frac{(-1)^k x^{2k+1}}{(2k+1)!}$	2	$\sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^{2k}}{k! 2^k - 1}$	3	$\sum_{k=1}^{\infty} \frac{(-1)^{k-1} x^k}{k!}$
4	$\sum_{k=1}^{\infty} \frac{(-1)^k x^{2k+1}}{k(2k+1)!}$	5	$\sum_{k=1}^{\infty} \frac{(-1)^k x^{2k-1}}{k(k+3)!}$	6	$\sum_{k=1}^{\infty} \frac{(-1)^k x^{2k+1}}{2k(k+1)!}$
7	$\sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^{2k}}{(2k-1)!}$	8	$\sum_{k=1}^{\infty} \frac{(-1)^{k-1} x^{3k-1}}{(2k)!}$	9	$\sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^{k+3}}{k^2(k+2)!}$
10	$\sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^{2k}}{(2k+1)!}$	11	$\sum_{k=1}^{\infty} \frac{(-1)^{k-1} x^{3k-1}}{(k+2)k!}$	12	$\sum_{k=1}^{\infty} \frac{(-1)^k x^{3k-1}}{(k+3)(3k)!}$
13	$\sum_{k=1}^{\infty} \frac{(-1)^{k-1} x^{3k+1}}{3k(k+1)!}$	14	$\sum_{k=1}^{\infty} \frac{(-1)^k x^{2(k+1)}}{(k+2)k!}$	15	$\sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^{3(k-2)}}{(k+3)(3k)!}$
16	$\sum_{k=1}^{\infty} \frac{(-1)^k x^{2k+1}}{2^k(2k-1)!}$	17	$\sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^k}{(k+4)!}$	18	$\sum_{k=1}^{\infty} \frac{(-1)^k x^{3k-1}}{(k+1)!k^2}$
19	$\sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^{2k}}{k! 2^k}$	20	$\sum_{k=1}^{\infty} \frac{(-1)^k x^{3k-1}}{2k(k+3)!}$	21	$\sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^{k+2}}{k(2k+1)!}$
22	$\sum_{k=1}^{\infty} \frac{(-1)^k x^{2(k+1)}}{(2k)!}$	23	$\sum_{k=1}^{\infty} \frac{(-1)^k x^{2k+1}}{k(2k+1)!}$	24	$\sum_{k=1}^{\infty} \frac{(-1)^{k-1} x^{3k-2}}{2k(k+3)!}$
25	$\sum_{k=1}^{\infty} \frac{(-1)^k x^k(2k-1)}{(3k-2)!}$	26	$\sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^{k-1}}{(2k-1)(k+1)!}$	27	$\sum_{k=1}^{\infty} \frac{(-1)^{k-1} x^{3k-2}}{2^{k+1}k!}$
28	$\sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^{k-1}}{(2+k)(2k+1)!}$	29	$\sum_{k=1}^{\infty} \frac{(-1)^{k-1} x^{3k+1}}{3k(2k-1)!}$	30	$\sum_{k=1}^{\infty} \frac{(-1)^k x^{2k-1}}{2k(2k+1)!}$

Таблиця 8.2

## Індивідуальні завдання базового рівня складності

№	Функція $f(x)$	№	Функція $f(x)$	№	Функція $f(x)$
1	$\sum_{k=1}^7 \frac{2^k \sin(x+k)}{(x+1)^k}$	2	$\sum_{k=1}^9 \frac{x^{k+1}}{(k+1)^x}$	3	$\sum_{k=1}^{12} \frac{\sin(kx) + k}{\sqrt[k]{x+0.1+6k}}$
4	$\sum_{k=1}^9 \frac{\ln(x+1)}{(x+k)^k}$	5	$\sum_{k=1}^9 \frac{\sin(2kx) + 0.2}{2k+5}$	6	$\sum_{k=1}^7 \frac{kx \cos(x+k)}{\ln(2+x) + 2k}$
7	$\sum_{k=2}^6 \frac{\sin(0.17x^k)}{2k+x}$	8	$\sum_{k=1}^8 \frac{5 \ln(2kx)}{\arctg(2x) + k^2}$	9	$\sum_{k=2}^9 \frac{\operatorname{tg}(x) - x^2/k}{k^2 - 1}$
10	$\sum_{k=1}^7 \frac{\sin(x^k - \pi)}{\ln k^2 + 0.3}$	11	$\sum_{k=1}^{12} \frac{\cos(kx)}{k}$	12	$\sum_{k=1}^8 \frac{\sin x^k}{4k}$
13	$\sum_{k=1}^7 \frac{\ln^k(3x)}{(2+x)^k}$	14	$\sum_{k=1}^8 \sqrt[k]{\ln(x+1)}$	15	$\sum_{k=6}^1 \frac{x^k}{k^3 + x^{k+2}}$
16	$\sum_{k=1}^{11} \frac{\sin(x^k - 1)}{4k^2 + 1}$	17	$\sum_{k=1}^8 \frac{\ln x^{2k-1}}{2^k(2k-1)}$	18	$\sum_{k=2}^9 \frac{\operatorname{tg}(e^x)}{3k^2 + 1}$
19	$\sum_{k=3}^{10} \frac{x^{k-1} \cos x}{12^k - 1}$	20	$\sum_{k=3}^{11} \frac{\cos^{2+k} x}{2k-1}$	21	$\sum_{k=1}^8 \sin^k(x)(k + \cos(x+2))$
22	$\sum_{k=2}^{10} \frac{\arctg^3(2kx)}{1.2 \ln(k+x)}$	23	$\sum_{k=1}^{11} \frac{\sin(x)^k + 0.3}{(2^k)}$	24	$\sum_{k=1}^6 \frac{k^2 \sin^2(x/k) - kx^2}{e^{kx}}$
25	$\sum_{k=1}^{12} \frac{\cos(x^k)}{(x+5)^k + k}$	26	$\sum_{k=2}^9 \frac{\sin(x+1) + 1.5}{\lg(5kx) + 2.1}$	27	$\sum_{k=1}^7 \frac{2(x+1)^{3-k}}{(k+1)^x + k^3}$
28	$\sum_{k=1}^{10} \cos\left(k^3 - \frac{kx}{5}\right)$	29	$\sum_{k=1}^7 \frac{x \sin(x-k)}{e^{2+x} + k}$	30	$\sum_{k=2}^6 x \cdot \arctg \frac{x - 4.4k}{x + \sin(x+k/5)}$

Таблиця 8.3

## Індивідуальні завдання високого рівня складності

№ вар.	Завдання
1	Ввести натуральне число та визначити суму цифр числа
2	Ввести натуральне число та визначити першу цифру числа, наприклад, для числа 3406 цифра 3
3	Ввести натуральне число та визначити суму всіх його дільників
4	Ввести натуральне число та визначити кількість парних цифр числа
5	Ввести натуральне число та визначити найбільшу цифру числа
6	Ввести натуральне число та змінити порядок цифр числа, наприклад, було 1234, стало 4321
7	Ввести натуральне число та визначити його цифри, які кратні 3

Закінчення табл. 8.3

№ вар.	Варіанти завдань
8	Ввести натуральне число та дописати в нього по 1 у початок і кінець
9	Ввести натуральне число та визначити скільки разів його перша цифра зустрічається у числі
10	Ввести натуральне число та визначити кількість його дільників менших 10
11	Ввести натуральне число та визначити чи є воно степенем числа 3
12	Ввести натуральне число та поміняти місцями першу й останню цифру числа, наприклад, з числа 1234 отримати 4231
13	Ввести натуральне число та дописати до нього таке само число, наприклад, з числа 1234 отримати 12341234
14	Ввести натуральне число та визначити кількість його парних дільників
15	Приписати до введенного числа таке само число, наприклад, з числа 1234 здобути 12341234
16	Ввести натуральне число та визначити кількість непарних цифр числа
17	Ввести натуральне число та визначити найменшу цифру числа
18	Ввести натуральне число та визначити його цифри, які кратні 5
19	Ввести натуральне число та визначити чи є воно степенем числа 2
20	Ввести натуральне число та визначити кількість його непарних дільників
21	Вивести всі двозначні числа, які діляться на 5 і містять цифру 5
22	Вивести всі трьохзначні числа, які при діленні на 47 дають в залишку 43, а при діленні на 43 дають в залишку 47
23	Ввести натуральне число та визначити кількість його дільників
24	Ввести натуральне число та визначити суму парних дільників
25	Вивести всі трьохзначні числа, кратні 7, і сума цифр яких також кратна 7
26	Вивести всі двозначні числа, які діляться на 9 або містять цифру 9
27	Вивести всі двозначні числа, сума квадратів цифр яких ділиться на 13
28	Ввести натуральне число та визначити суму непарних дільників
29	Ввести натуральне число та поміняти місцями першу і другу цифру числа, наприклад, з числа 1234 отримати 2134
30	Ввести натуральне число та продублювати кожну з цифр числа двічі, наприклад, з числа 1234 отримати 11223344