Лабораторна робота з дисципліни «Алгоритмізація та програмування» № 4

Тема: Розробка програм з використанням циклів

Мета роботи: Вивчення правил побудови та алгоритмів роботи операторів мови С для організації циклічних обчислень.

Теоретичні відомості

Оператор циклу з параметром

Цикл *for* переважно використовується, коли наперед відома кількість повторень, або коли умова продовження виконання циклу записується коротким виразом.

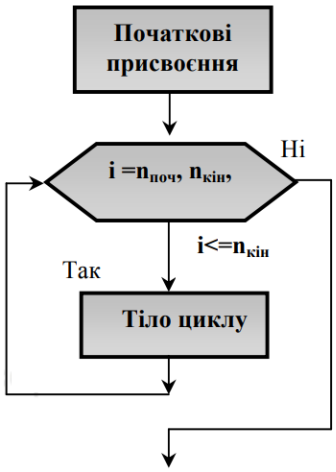
Оператор циклу for має такий формат:

```
for (<ініціалізація>; <умова>; <модифікація>)
    <тіло_циклу>;
```

За замовчуванням після оператора for виконується лише один оператор, тому якщо у циклі потрібно виконати групу команд, то їх слід записати в операторних дужках {}.

```
for (<ініціалізація>; <умова>; <модифікація>)
{
    <оператор_1>;
    <оператор_2>;
    ...
    <оператор_N>; }
```

Для зображення циклу на блок-схемі використовується наступна сукупність блоків:



Оператор for складається з трьох основних блоків, записаних у круглих дужках і відокремлених один від одного крапкою з комою (;), та тіла циклу.

У блоці ініціалізації задаються початкові значення змінних (параметрів), які керують циклом. У блоці умови задається та перевіряється умова і, якщо вона виконується (true чи має ненульове значення), то виконується тіло циклу. Якщо ж умова не виконується (false чи дорівнює нулю), то відбувається вихід із циклу, а керування передається на перший оператор після циклу for.

Перевірка умови виконується на початку кожної ітерації циклу.

Для обчислення суми перших десяти чисел можна скористатись наступними конструкціями

```
int s = 0;
for (int i = 1; i <= 10; i++)
    s += i;
```

або

```
for (int s = 0, i = 10; i >= 1; i--)
    s += i;
```

або

```
for (int s = 0, i = 1; i <= 10; s += i++);
```

В операторі for блок ініціалізації може бути відсутнім, якщо початкове значення задати попередньо; блок умови також, якщо припускається, що умова завжди істинна (слід виконувати тіло циклу, поки не зустрінеться оператор break) а блок модифікації може бути пропущений, якщо зміну значення параметра виконувати у тілі циклу чи коли значення параметра змінювати непотрібно.

Коли певний блок відсутній, тоді вираз цього блока пропускається, але крапка з комою (;) обов'язково має залишитись. Крім того, можлива наявність порожнього оператора (оператор відсутній) у тілі циклу.

При використанні вкладених циклів потрібно слідкувати, щоб внутрішній цикл повністю вкладався у тіло зовнішнього циклу. Крім того, внутрішній цикл може містити власні вкладені цикли. Імена параметрів зовнішнього та внутрішнього циклів мають бути різними.

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

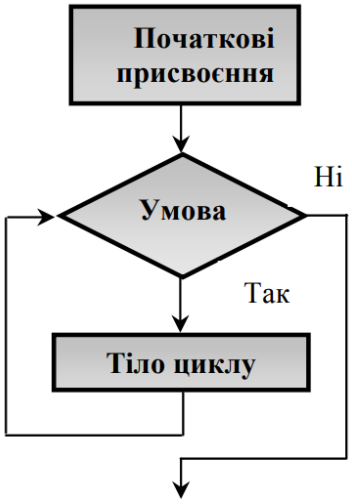
Оператор циклу з передумовою

Цикл з передумовою *while* використовується, якщо кількість повторень наперед невідома або ж немає явно вираженого кроку зміни параметра циклу.

Синтаксис циклу з передумовою:

while (<умова>)
{ <тіло_циклу> };

Для зображення циклу на блок-схемі використовується наступна сукупність блоків:



Умова цикла *while* перевіряється перед кожною ітерацією. Тіло циклу виконується доти, поки умова істинна (true, має ненульове значення), а вихід з цикла здійснюється, коли умова стає хибною (false, має нульове значення). Якщо умова є хибною при входженні у цикл, то послідовність операторів не буде виконуватися жодного разу, а керування передасться наступному оператору програми.

Для обчислення суми всіх непарних чисел у діапазоні від 10 до 100 можна використати наступний цикл:

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

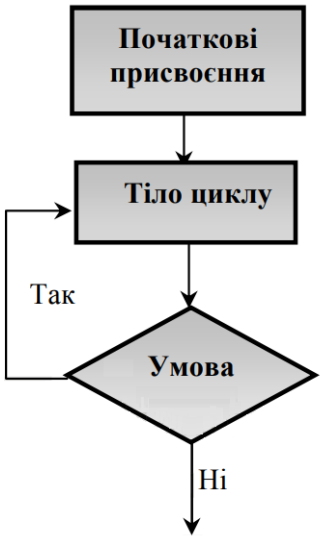
Оператор циклу з післяумовою

Відмінність циклу з післяумовою *do - while* від циклу з передумовою полягає в тому, що цикл з післяумовою завжди виконується принаймні один раз незалежно від істинності чи хибності умови.

Синтаксис циклу з післяумовою:

```
do {
  <тіло_циклу> }
 while (<умова>);
```

Для зображення циклу на блок-схемі використовується наступна сукупність блоків:



Тіло циклу виконується доти, поки умова є істинна (ненульова). Якщо тіло циклу складається з одного оператора, то операторні дужки {} не обов'язкові.

Обчислити програмно суму всіх непарних чисел у діапазоні від 10 до 100 використовуючи цикл з післяумовою можна так:

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

У тілі циклів з передумовою та післяумовою слід передбачати зміну параметрів, які використовуються в умові, інакше вихід із циклу ніколи не виконається і відбуватиметься зациклювання.

Оператори переривання виконання циклів

Для завчасного виходу з циклу використовують оператори *break* (вихід з конструкції), *goto* (безумовний перехід) чи *return* (вихід з поточної функції).

Оператор break перериває виконання оператора, в якому він розміщений, а керування передається на наступний оператор:

```
 int k = 1;
for (int i = 0; i < m - 1; i++)
{
    for (int j = i + 1; j < m; j++)
        if (i == m - j) break;
        if (j == m) k++;
}
```

Всередині вкладених операторів do-while, for, while чи switch оператор break завершує лише той оператор, якому він належить, тому break неможна використовувати для виходу з декількох вкладених циклів. Навіть подвійне послідовне використання двох операторів break не забезпечить вихід із вкладених циклів.

```
for (i = 0; i < 100; i++)
    for (j = 0; j < 100; j++)
     {
...
        if (j - i) < 0) { break; break; }
...
          }
```

Після виконання умови (j-i)<0 відбудеться вихід лише з внутрішнього циклу по змінній j, а виконання зовнішнього циклу по змінній i продовжиться, незважаючи на те, що оператор break записано двічі.

Оператор break є оператором переходу і оператори записані після нього виконуватись не будуть, якщо тільки їм не буде передано керування за допомогою інших операторів переходів.

Для переходу до наступної ітерації циклу призначений оператор *continue,* який переходить до наступної ітерації відповідного циклу і використовується лише всередині операторів циклів for, while, do-while.

```
for (int i = 3; i < 7; i++)
    if (i == 5) continue; // Пропустити елемент, рівний 5
    else
              cout << i; // Результат: 3  4  6  7
```

Крім того, передавати керування за межі вкладеної структури, можна використовуючи оператори return та goto.

Завдання 1

1. Написати програму обчислення виразу (суми чи добутку послідовності) згідно свого варіанта. Передбачити перевірку введення правильних даних. Передбачити перевірку введення правильних даних. Для введення/виведення даних використати функції потокового введення-виведення cin() та cout().
2. Побудувати блок-схему.

3. Введення та виведення даних необхідно супроводжувати відповідними текстовими повідомленнями.

Варіант 1

$$S = \sum_{i=1}^{10} \frac{(-1)^i i^2}{i^2 + 5i + 6}$$

Варіант 2

$$S = \prod_{i=1}^{14} \frac{(-1)^{i+4}(5i^2 + i)}{6i^2 + i + 3}$$

Варіант 3

$$S = \sum_{i=1}^{15} \frac{(-1)^{i+5}(6i^2 + 1)}{7i^2 + 2i + 4}$$

Варіант 4

$$S = \sum_{i=1}^{18} \frac{(-1)^{i+8}(6i^2 + 7)}{5i^2 + 6i + 5}$$

Варіант 5

$$S = \sum_{i=1}^{12} \frac{(-1)^{i+2} i^2 + 4}{i^2 + 8i + 15}$$

Варіант 6

$$S = \prod_{i=1}^{11} \frac{(-1)^{i+1}(i^2 + 2i + 5)}{i^2 + 2i}$$

Варіант 7

$$S = \sum_{i=1}^{10} \frac{(-1)^i (i^2 + 3)}{3i^2 + 2i + 2}$$

Варіант 8

$$S = \sum_{i=1}^{13} \frac{(-1)^{i+3}(i^2 + 9)}{i^2 + 9i + 1}$$

Варіант 9

$$S = \prod_{i=1}^{11} \frac{(-1)^{i+1}(2i^2 + 3i)}{i^2 + 3i + 4}$$

Варіант 10

$$S = \prod_{i=1}^{10} \frac{(-1)^i (i^2 + 4i)}{i^2 + 4i + 5}$$

Варіант 11

$$S = \sum_{i=1}^{11} \frac{(-1)^{i+1}(i+1)}{i^2+2i+7}$$

Варіант 12

$$S = \sum_{i=1}^{14} \frac{(-1)^{i+4}}{5i^2+i+2}$$

Варіант 13

$$S = \sum_{i=1}^{17} \frac{(-1)^{i+7}(4i^2+1)}{3i^2+4i+3}$$

Варіант 14

$$S = \prod_{i=1}^{13} \frac{(-1)^{1+4}(i^2+3i)}{2i^2+3i+4}$$

Варіант 15

$$S = \prod_{i=1}^{16} \frac{(-1)^{i+6}(3i^2+i)}{2i^2+3i+2}$$

Варіант 16

$$S = \sum_{i=1}^{13} \frac{(-1)^{i+3}(3i^2+4)}{2i^2+2i+1}$$

Варіант 17

$$S = \sum_{i=1}^{16} \frac{(-1)^{i+6}(7i^2+4)}{5i^2+3}$$

Варіант 18

$$S = \prod_{i=1}^{12} \frac{(-1)^{i+2}(i+2)}{i^2+5i+3}$$

Варіант 19

$$S = \sum_{i=1}^{14} \frac{(-1)^{i+4}(6i^2+1)}{5i^2+4}$$

Варіант 20

$$S = \sum_{i=1}^{13} \frac{(-1)^{i+3}(i^2+3)}{i^2+7i+4}$$

Завдання 2

1. Протабулювати задану функцію на заданому інтервалі двома способами (з використанням двох різних операторів циклу) в одній програмі. Передбачити

перевірку введення правильних даних. Для введення-виведення даних використати функції потокового введення-виведення cin() та cout().

2. Побудувати блок-схему.

3. Перед виведенням результатів певного циклу вивести повідомлення про цикл, який використовується. Результати всіх циклів мають співпадати. Введення та виведення даних необхідно супроводжувати відповідними текстовими повідомленнями.

№	$y=f(x)$	$x_{поч}$	$x_{кін}$	h	a	b
1	$y = \frac{e^{\sin x} + \sqrt[4]{a+x}}{\ln^3 bx}$	5.5	10.5	0.5	17.3	0.36
2	$y = \frac{\ln^4 bx + 0.85}{\sqrt[3]{a+bx^3}}$	0.4	6.8	0.8	46	1.85
3	$y = \frac{\sqrt[4]{1+\sqrt{ax+b}}}{sin^2 bx + a}$	4.3	13.9	1.2	1.35	8.4
4	$y = \frac{tg^2(ax) + \sqrt{\ln x}}{e^{-bx}}$	1.3	6.1	0.6	1.8	0.56
5	$y = \frac{\sqrt[3]{a^3+x^3}}{tg^3 bx + 1.6}$	0.2	1.6	0.2	1.25	0.86
6	$y = \frac{\sqrt[3]{ax}+b}{0.25\ln^2 ax}$	10.5	28.5	2	0.3	9.5
7	$y = \frac{ln^2\,(a^3+x^3)}{\sqrt{a^3+x^3}+\sqrt[3]{b}}$	8.2	98.2	10	43	205
8	$y = \frac{1+\cos^2(a^2+x^2)}{x^3+\sqrt[3]{bx}}$	0.5	1.9	0.2	0.84	0.63
9	$y = \frac{a^x + e^{-bx}}{\sin^2 bx + 1.24}$	0.3	1.3	0.1	0.5	0.16

10	$y = \sqrt{\dfrac{\lvert bx \rvert}{arctg\,\dfrac{b^2}{a^2+x^2}}}$	-10	-1	1	2.8	1.5
11	$y = \dfrac{(x^2+1) - \dfrac{1}{\sin bx}}{\sqrt[3]{\dfrac{x}{a}} - 0.39}$	0.2	1.6	0.1	0.36	0.74
12	$y = \dfrac{e^{x^2+1}}{\sqrt[5]{\lvert x-a \rvert} + ln^2 bx}$	1.2	3	0.2	4.8	6.8
13	$y = \dfrac{e^{sin^2 ax} + arctgbx}{\sqrt[3]{(x+b)^2}}$	0.5	14.5	2	0.45	8.8
14	$y = \dfrac{sin^2 ax + \sqrt[3]{\lvert x-b \rvert}}{\lvert x-b \rvert^3}$	16	22	0.5	0.28	19.3
15	$y = \dfrac{x^{\frac{a}{b}} - \sqrt[3]{\dfrac{x+b}{a}}}{1.1 + cos^2 ax}$	6.8	20.8	1	3.5	6.4
16	$y = \sqrt[5]{\dfrac{x^2}{4a^2+0.6}}\, ln^2(b+x)$	10	20	1	2.4	16.8
17	$y = \dfrac{cos^2 ax + e^{-ax}}{arctg(\sqrt{b} + \sqrt{x})}$	0.6	0.8	0.02	1.3	0.75
18	$y = \dfrac{a - e^{bx}}{(x-a)^2 + ln^2(x-a)}$	2	8.5	0.5	4.38	0.24
19	$y = \dfrac{\sqrt[5]{a+x} - b}{1.6\, cos^{3.2} ax}$	0.3	3.3	0.3	1.85	2.64
20	$y = \dfrac{e^{a+x}}{\sqrt[4]{bx} + ln^2 ax}$	0.8	2.4	0.2	1.38	9.6