

Міністерство освіти і науки України
Харківський національний автомобільно-дорожній університет
Кафедра інформаційних технологій та мехатроніки

МЕТОДИЧНІ ВКАЗІВКИ
до лабораторних робіт з дисципліни
“Інформаційні технології”

**“Програмування на мові C++
у середовищі Microsoft Visual Studio 2010”**

для студентів напряму підготовки 6.050702 ” Електромеханіка ”,
галузь знань 0507 ” Електротехніка та електромеханіка ”

Розроблено та надруковано доц. Симбірським Г.Д.

Харків, 2015

Лабораторная работа № 4 РАЗРАБОТКА И ИССЛЕДОВАНИЕ ЦИКЛИЧЕСКИХ ПРОГРАММ В VISUAL C++ 2010 (ОПЕРАТОР ЦИКЛА FOR)

Цель работы: получение практических навыков в разработке циклических программ с использованием оператора цикла **for**.

1. Циклические вычислительные процессы

Возможность повторно выполнять некоторые действия очень важна при разработке любых программ и программных приложений. Вычислительный процесс, содержащий многократные вычисления по одним и тем же математическим зависимостям, называется **циклическим**.

Цикл выполняет оператор или группу операторов до тех пор, пока истинно (или ложно) определенное условие относительно некоторой переменной, называемой **параметром цикла**.

Многократно повторяющиеся части такого процесса составляют **тело цикла**.

Алгоритм циклических структур должен содержать (рис. 4.1):

1. **Подготовку к циклу** – присваивание начального значения параметру цикла.
2. **Проверку условия** выполнения тела цикла.
3. **Тело цикла** – действия, которые выполняются в циклической программе для разных значений параметра цикла.
4. **Изменение (модификация) значений параметра цикла**.

На рис. 4.1 изображена блок-схема алгоритма циклического вычислительного процесса, где помимо характеристик операционных блоков в качестве примера приведены реальные операторы.

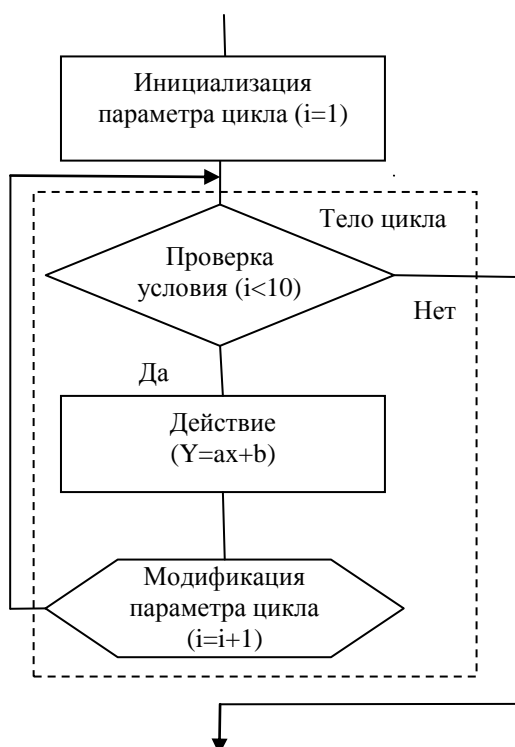


Рис. 4.1. Блок-схема алгоритма циклического вычислительного процесса

В среде Visual C++ 2010 циклические вычислительные процессы реализуются с помощью операторов **while**, **do...while** и **for**. Выше были исследованы операторы **while** и **do...while**. В настоящей работе будут исследованы программы с использованием оператора **for**.

2. Оператор цикла for

Оператор цикла **for** используется, когда количество повторений тела цикла **заранее известно**. Форма записи оператора цикла **for** следующая:

for ([выражение инициализации]; [выражение проверки (условие)]; [выражение модификации])
оператор внутри цикла;

Квадратные скобки показывают, что данная секция в операторе может быть опущена.

На практике это выглядит, например, следующим образом:

```
for(i=1; i<=n; i=i+1)  
Y = a*i;
```

где **i** – параметр цикла.

Анализ данной записи показывает, что оператор **for** объединяет в себе три операционных блока из блок-схемы циклического вычислительного процесса (рис. 4.1):

- блок инициализации, т. е. присвоения параметру цикла начального значения (**i=1**);

- блок проверки условия ($i \leq n$);
- блок модификации параметра цикла ($i=i+1$).

Это свойство оператора цикла **for** позволяет существенно упростить вычислительные процессы и программные коды при решении различных задач в Visual C++ 2010.

В схемах алгоритмов оператор цикла **for** отражается символом **модификация** (рис. 4.2):

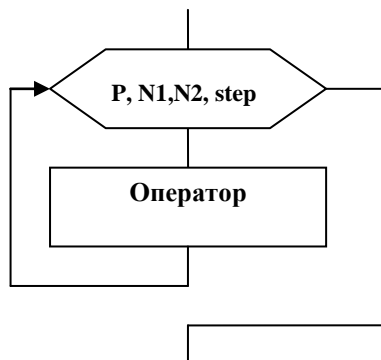


Рис. 4.2. Блок-схема оператора цикла **for**

На схеме алгоритма приведены следующие обозначения:

- **P** – параметр цикла;
- **N1, N2** – границы изменения параметра цикла;
- **step** – шаг изменения параметра цикла (если шаг не указан, то он равняется 1) .

Параметр цикла **P**, границы изменения **N1, N2** и шаг **step** должны иметь один и тот же тип.

В языке C++ принято операцию инкремента (приращения цикла) $i=i+1$, записывать как $i++$, например:

```
for(i=1; i<=n; i++)
    f=f*i;
```

Возможности оператора цикла **for** очень велики. Например, вместо любого из трех выражений в записи общей формы можно записать два и более выражения, разделенных запятыми:

```
for(i=1, j=1, z=1; i<=n; i++, j++, z++)
    f=f*i*j*z;
```

Рассмотрим простейший пример оператора цикла **for**.

Задание 4.1. Исследовать программу для печати чисел от 1 до заданного числа $n=15$ с шагом 1. Использовать оператор **for**. Блок-схема алгоритма выполнения такого задания приведена на рис. 4.3.

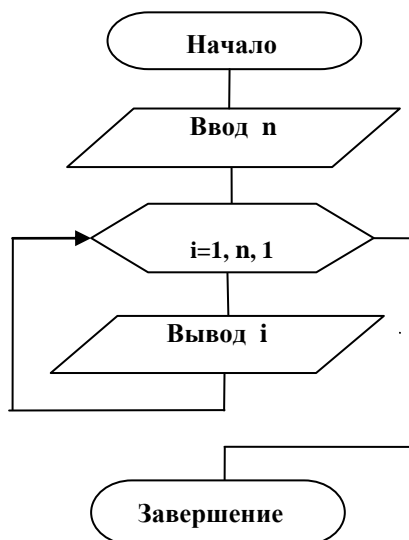


Рис. 4.3. Блок-схема алгоритма для печати в столбец **n** чисел

Ниже приведен программный код, реализующий данный алгоритм:

```
#include "stdafx.h"
#include <conio.h>
#include "iostream"
using namespace std;
```

```
int _tmain(int argc, _TCHAR* argv[])
```

```

{
int i, n ;
cout<<"Vvedite n "<<endl;
cin>>n;                               //Ввод значения переменной n
for ( i=1; i<=n; i++)                 //Оператор цикла for
    cout << i<<" ";                   //Тело цикла – вывод параметра цикла i
getch();
return 0;
}

```

В результате выполнения программы будут напечатаны положительные числа от 1 до 15:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15.

Анализ результатов показывает, что работа оператора цикла **for** происходит в полном соответствии со схемой вычислительного процесса на рис. 4.1. В данном примере параметром цикла **for** является переменная **i** (она же – счетчик). В начале цикла счетчик (переменная **i**) инициализируется значением 1. Потом выполняется тело цикла (**cout << i << endl;**) и проверяется, не достиг ли счетчик значения **n=15**. После каждого выполнения тела цикла счетчик (**i**) увеличивается на единицу. Как только **i** станет равным 15, тело цикла пропускается и управление передается следующему оператору программы.

Создайте в текстовом процессоре **Word** файл **Результат_Фамилия_Лр6**. Поля документа сделайте по 0,5 см.

Поместите окно DOS с результатами решения **Задания 4.1** в центральной части окна **Microsoft Visual Studio** ниже программного кода **Lr4-1.cpp** (см. рис. 1.2) и нажмите клавишу **<Prt Scr>**, после чего вставьте полученную копию экрана в файл **Результат_Фамилия_Лр4**. Над вставленным рисунком проставьте номер задания – **4-1**. Файл результатов не закрывайте до получения оценки за выполненную практическую часть работы в тетрадь с отчетом.

Закройте окно DOS, откройте пункт меню **Файл** и выполните команду **Закрывать решение**.

!Внимание! Результаты следующих заданий данной лабораторной работы сохраняйте строго в соответствии с приведенным выше порядком действий!

Задание 4.2. Исследовать программу вычисления факториала целого числа **n** с использованием оператора **for**. **Факториал** – это произведение целых чисел от 1 до **n**. Необходимо проанализировать блок-схемы вычислительного процесса и алгоритма решения задания при помощи оператора цикла **for**, ввести код программы, построить решение и произвести вычисления для **n=10**.

Факториал числа **n** вычисляется по следующей формуле:

$$n! = 1*2*3*4* \dots *n = \prod_{i=1}^n i .$$

При вычислении факториала начальному значению произведения Π необходимо присвоить 1. При каждом выполнении тела цикла Π_{i-1} будем умножать на **i**. Примем для обозначения произведения идентификатор **p**. Блок-схемы вычислительного процесса и алгоритма решения задания при помощи оператора цикла **for** имеют следующий вид:

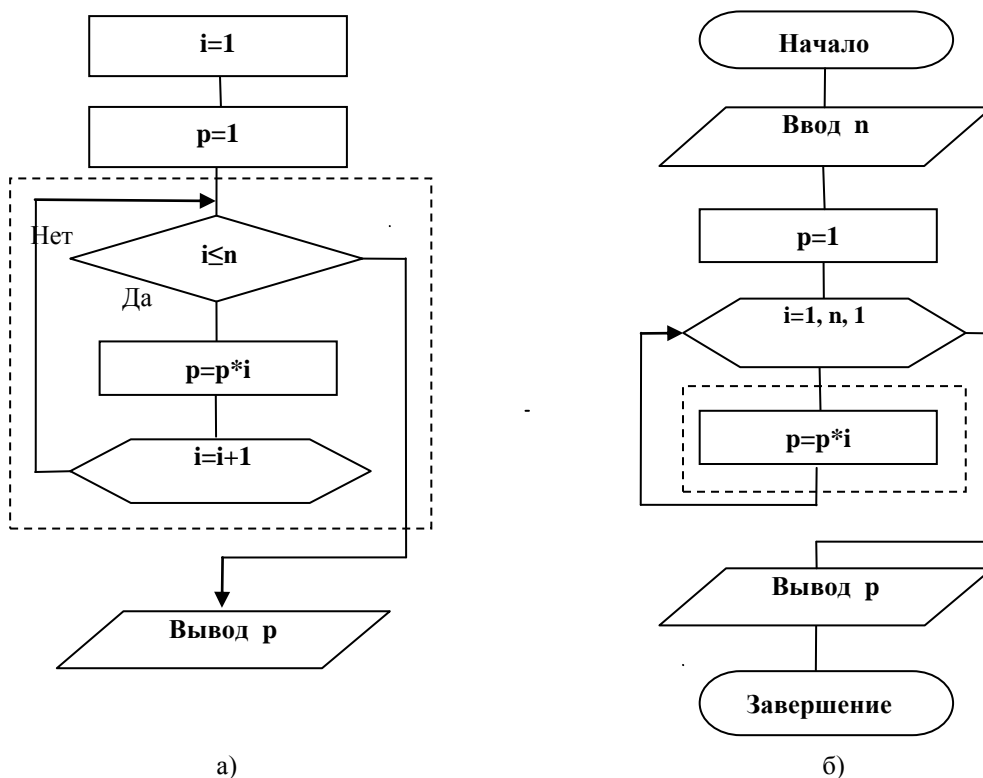


Рис. 4.4. Блок-схемы вычисления факториала целого числа **n**:
а) вычислительного процесса; б) алгоритма с использованием оператора **for**

Программу вычисления факториала можно записать в следующем виде:

```
#include "stdafx.h"
#include <conio.h>
#include "iostream"
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
    int i, n, p;
    cout<<"Vvedite N"<<endl;
    cin>>n; //Ввод значения переменной n
    p = 1; //Начальное значение 1 переменной p для произведения
    for (i=1; i<=n; i++) //Оператор цикла for
        p = p*i; //Тело цикла – произведение p и i
    cout<<"faktorial " <<n<<" = "<<p<<endl; //Печать результата
    getch();
    return 0;
}
```

После запуска программы на экране появится результат:

```
Vvedite celoe chislo: 5
faktorial 5 = 120
```

Обратите внимание на необходимость присваивания начального значения (единица) переменной **p** для вычисления последующих произведений.

Задание 4.3. Исследовать программу для вычисления суммы **n** четных чисел, начиная от двух. Использовать оператор **for**. Формула для вычисления такой суммы имеет следующий вид:

$$s = \sum_{i=1}^n (2 * i),$$

где **i** – параметр цикла (номер четного числа на данном шаге).

Блок-схема алгоритма решения такой задачи приведена на рис. 4.5. Обратите внимание на необходимость присваивания начального значения (ноль) переменной **s** для вычисления последующих сумм.

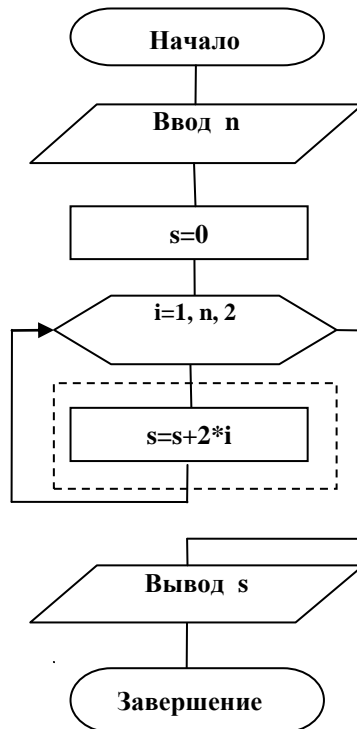


Рис. 4.5. Блок-схема алгоритма для вычисления суммы **n** четных чисел

Программу вычисления суммы **n** четных чисел можно записать в следующем виде

```

#include "stdafx.h"
#include <conio.h>
#include "iostream"
using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
    int i, n, s;
    cout<<"Vvedite N"<<endl;
    cin>>n; //Ввод значения переменной n
    s=0; //Начальное значение 0 переменной s для суммирования
    for ( i=1; i<=n; i++) //Оператор цикла for
        s=s+2*i; //Тело цикла – расчет суммы
    cout<<"Summa " <<2*n<<" = "<<s<<endl; //Печать результата
    getch();
    return 0;
}

```

После запуска программы на экране появится результат:

```

Vvedite celoe chislo: 5
Summa 10 = 30

```

Задание 4.4. Исследовать программу вычисления функции $Y = \sum_{k=1}^6 \left(\cos^3 kx + \sin \frac{0,5x}{\sqrt[3]{k^2 + x^2 + 1}} \right)$ для $x=1,32$. Блок-

схема алгоритма для вычисления данной функции аналогична блок-схеме алгоритма для вычисления суммы n четных чисел на рис. 4.5. Необходимо ввести код программы, построить решение и произвести вычисления.

Для решения задачи используется оператор цикла **for**, т. к. известно количество повторений тела цикла ($k=10$). Введем такие идентификаторы: x , Y , k и S . Формула для расчета функции Y на каждом шаге вычислений на языке C++ имеет следующий вид:

$$Y = \text{pow}(\cos(k*x), 3) + \sin(0.5*x / \text{pow}((k*k + x*x + 1), 1/3)).$$

Тогда программа для расчета функции Y примет вид:

```

#include "stdafx.h"
#include <conio.h>
#include "iostream"
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
    int k;
    double x=1.32, Y, S;
    Y=0; //Начальное значение переменной Y для суммирования
    cout<<"x= "<<x<<endl; //Вывод x
    for (k=1; k<=6; k++) //Оператор цикла for
    { //Начало тела цикла
        S=pow(cos(k*x),3)+sin(0.5*x/pow((k*k+x*x+1),1/3)); //Расчет функции на каждом шаге
        Y=Y+S;
        cout<<"k= "<<k<<" Y = "<<Y<<endl; //Вывод суммы Y на каждом шаге
    } //Конец тела цикла
    getch();
    return 0;
}

```

Так как тело цикла содержит более одного оператора, то эти операторы охвачены фигурными скобками.

После запуска программы на экране появится результат:

```

x = 1.32
k = 1    Y = 1.1365
k = 2    Y = 2.27299
k = 3    Y = 3.40949
k = 4    Y = 4.54599
k = 5    Y = 5.68249
k = 6    Y = 6.81898
k = 7    Y = 7.95548
k = 8    Y = 9.09198
k = 9    Y = 10.2285

```

Рис.4.6. Результаты вычислений для Задания 4.4

Обратите внимание на то, что результат расчета функции Y выводится на каждом шаге вычислений. Это сделано с целью контроля при отладке программы. После устранения возможных ошибок можно выводить только итоговый результат.

Задание 4.5. Самостоятельно разработать алгоритм и программу для вычисления значения функции Y с использованием оператора **for** (таб. 4.1). Начертить в отчете блок-схему алгоритма решения задачи и запишите формулу для расчета Y . Определите свой номер варианта как номер компьютера.

Таблица 4.1 Исходные данные и формулы для расчета Y (Задание 4.5)

№ варианта	Формула для расчета Y	Значения x, a
1	$Y = \sum_{k=1}^8 \frac{\sin^3(x+a) - k \cos^2(x+a)}{(x+a)^4}$	$x=1,7; a=1,77$
2	$Y = \prod_{k=0}^6 \frac{\operatorname{tg}^3(x+a) - \arccos^2(x+a)}{k(x+a)^4}$	$x=2,7; a=1,33$
3	$Y = \sum_{k=0}^5 \left(\operatorname{ctg} \frac{1-3x}{1+2x} + \cos^2 5x + ke^{3a} \right)$	$x=0,7; a=0,46$
4	$Y = \prod_{k=0}^8 \frac{\cos^3(x+a) - k7(x+a)}{\operatorname{tg}(x+a)^4}$	$x=2,7; a=1,82$
5	$Y = \sum_{k=1}^7 \frac{\cos\left(k \frac{3a+1}{4}\right)}{\sin^3 3x + e^{4a}}$	$x=0,45; a=0,82$
6	$Y = \prod_{k=1}^6 \frac{\sin^3(x+a) - \cos^2(x+a)}{k(x+a)^4}$	$x=2,1; a=1,47$
7	$Y = \sum_{k=0}^5 \frac{\operatorname{tg} \frac{4a^2+1}{4}}{\cos^3 2x + ke^{2a}}$	$x=2,1; a=1,34$
7	$Y = \prod_{k=0}^8 \left(\sin \frac{1-x}{1+x} + k \operatorname{tg}^4 5x + e^{5a} \right)$	$x=0,7; a=1,28$
9	$Y = \sum_{k=1}^5 \frac{\sin^3(x+a) - k \arccos^2(x+a)}{\cos(x+a)^4}$	$x=2,2; a=0,66$
10	$Y = \prod_{k=0}^7 \frac{\operatorname{ctg}\left(k \frac{x^3+1}{4}\right)}{\cos^2 5x + e^{3a}}$	$x=1,45; a=1,12$
11	$Y = \sum_{k=1}^6 \frac{\operatorname{ctg}^3(3x+a) - k \sin^2(x+7a)}{(5x+a)^3}$	$x=2,7; a=1,82$
12	$Y = \prod_{k=1}^4 \frac{\arcsin^3 \frac{4x+1}{4}}{\operatorname{ctg}^2 k(3x+e^{3a})}$	$x=1,25; a=1,42$
13	$Y = \sum_{k=0}^6 \frac{\sin^3 \frac{3x+1}{2}}{\operatorname{tg}^2 5x + ke^{3a}}$	$x=1,85; a=1,72$
14	$Y = \prod_{k=1}^8 \frac{\sin^3(x+a) - \cos^2(x+a)}{k(x+a)^4}$	$x=1,48; a=1,19$

14	$Y = \sum_{k=0}^7 \frac{\operatorname{arctg} \frac{2kx^3 + 1}{4}}{\cos^2 5x + e^{3a}}$	$x=1,15; a=0,12$
16	$Y = \prod_{k=1}^7 \frac{\operatorname{tg}^3(x+a) - 5k(\sin x + a)}{\sin^3(x+a)^4}$	$x=2,45; a=2,12$
17	$Y = \sum_{k=0}^6 \left(k \times \operatorname{tg} \frac{1-x}{1+x} + k \times \sin^2 5x + e^{5a} \right)$	$x=2,25; a=1,88$
18	$Y = \prod_{k=1}^8 \frac{\sin \frac{x+1}{4}}{\sin^2 5x + ke^{3a}}$	$x=2,4; a=0,56$

Задание 4.6. Исследовать программу вычисления значения функции $Y = \sum_{n=0}^4 n \left(\sum_{k=1}^5 \sin^3(knx - a) \right)$ для $x=1,4$ и $a=2,3$ (использовать цикл в цикле с оператором **for**). Блок-схема алгоритма решения данной задачи приведена на рис. 4.7. Необходимо ввести код программы, построить решение и произвести вычисления.

Определим исходные данные, которые понадобятся для решения задачи:

a – константа, инициализируем ее, как $a = 2,3$ и определим ее тип как **double** (действительное двойной точности);

x – константа, инициализируем ее, как $x=1,4$, тип **double**;

Y – переменная, тип **double**;

s – промежуточная переменная, обозначим ее как **S**, тип **double**.

Формула для расчета промежуточной суммы (внутри цикла по **k**) $s = \sin^3(knx - a)$ на **k**-м шаге вычислений на языке C++ имеет следующий вид:

$$s = \operatorname{pow}(\sin(k*n*x - a), 3) .$$

Блок-схема алгоритма решения данной задачи приведена на рис. 4.6.

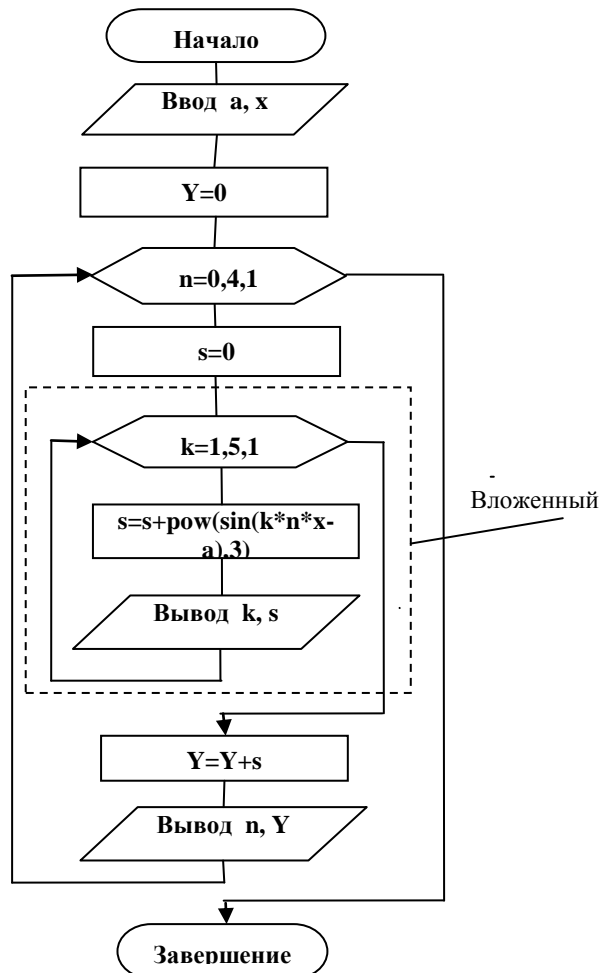


Рис. 4.7. Блок-схема алгоритма решения задания 4.6 с использованием вложенного цикла **for**

Введите и выполните программный код, реализующий данный алгоритм. Обратите внимание на сдвиг фигурных скобок, ограничивающих сложные операторы, относительно скобок, охватывающих тело главной функции.

```

#include "stdafx.h"
#include <conio.h>
#include <iostream>
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
    const double x=1.4, a=2.3;
    double Y, n, k, s;
    cout<<"x= "<<x<<" a= "<<a<<endl;
    Y=0;
    for(n=0;n<=4;n++)
    {
        s=0;
        for(k=1;k<=5;k++)
        {
            s=s+ pow(sin(k*n*x-a),3);
            cout<<k<<" s = "<<s<<endl;
        }
        Y=Y+n*s;
        cout<<n<<" Y = "<<Y<<endl;
    }
    getch();
    return 0;
}

```

//Файл **conio.h** обеспечивает задержку окна DOS на экране
//Директива **include** подключает файл ввода-вывода **iostream**
//Подключает все имена из пространства имен **std**
//Объявление главной функции **_tmain**
//Инициализация констант **a** и **x**
//Объявление переменных и параметров цикла: **y, s, n** и **k**
//Вывод исходных данных
//Начальное значение переменной **Y** для суммирования
//Внешний оператор **for** (условие цикла **n ≤ 4**)
//Начало составного оператора – вложенного цикла
//Начальное значение переменной **s** для суммирования
//Вложенный оператор **for** (условие цикла **k ≤ 5**)
//Суммирование **s** на **k**-ом шаге
//Вывод на экран переменной **y** на **k**-ом шаге
//Суммирование **Y** на **n**-ом шаге
//Вывод на экран переменной **s** на **k**-ом шаге
//Конец составного оператора с вложенным циклом
//Функция задержки окна DOS на экране
//Конец главной функции

Результаты вычислений приведены на рис. 4.7. Промежуточные результаты вычислений выводятся на экран для дополнительного контроля правильности работы программы (тестирования программы).

```

x= 1.4  a= 2.3
k = 1   s = -0.414669
k = 2   s = -0.829338
k = 3   s = -1.24401
k = 4   s = -1.65868
k = 5   s = -2.07334

      n = 0  Y = 0
k = 1   s = -0.48065
k = 2   s = -0.370455
k = 3   s = 0.476942
k = 4   s = 0.473016
k = 5   s = -0.526753

      n = 1  Y = -0.526753
k = 1   s = 0.110195
k = 2   s = 0.10627
k = 3   s = 0.100225
k = 4   s = 0.225993
k = 5   s = -0.21643

      n = 2  Y = -0.959613
k = 1   s = 0.847396
k = 2   s = 0.841352
k = 3   s = 0.388923
k = 4   s = 1.20605
k = 5   s = 1.20274

      n = 3  Y = 2.64861
k = 1   s = -0.0039253
k = 2   s = 0.121842
k = 3   s = 0.938967
k = 4   s = 1.79397
k = 5   s = 1.94911

      n = 4  Y = 10.445

```

Рис. 4.8. Результаты вычислений для Задания 4.6

Задание 4.7. Самостоятельно создайте проект для вычисления функции **Y** по заданной формуле в соответствии со своим вариантом (номер компьютера). Варианты заданий находятся в таблице 4.2. Ввод исходных данных и вычисления организуйте по аналогии с **Заданием 4.6**.

Перед (!) разработкой программного кода начертите в отчете блок-схему алгоритма решения и запишите формулу для вычислений на языке C++.

Таблица 4.2. Исходные данные и формулы для расчета Y (Задание 4.7)

№ варианта	Формула для расчета Y	Значения a и x
1	$s = \sum_{n=1}^3 \sum_{k=1}^6 \sqrt[3]{(knx^2 - an)}$	a=1,2; x=1,32
2	$s = \sum_{n=1}^4 \sum_{k=1}^5 \operatorname{tg}^4(knx^3 - an)$	a=1,3; x=1,46
3	$s = \sum_{n=1}^6 \sum_{k=1}^5 \sin^4(knx^3 - a)$	a=1,4; x=1,7
4	$s = \sum_{n=1}^6 \sum_{k=1}^3 \sqrt[4]{(knx^3 - ak)}$	a=1,5; x=1,4
5	$s = \sum_{n=1}^4 \sum_{k=1}^3 \operatorname{tg}^3(knx^5 - a)$	a=1,1; x=1,6
6	$s = \sum_{n=1}^5 \sum_{k=1}^4 \sqrt[3]{(knx^2 - a)}$	a=1,2; x=1,8
7	$s = \sum_{n=1}^4 \sum_{k=1}^3 \operatorname{tg}^5(knx^3 - an)$	a=1,3; x=1,4
8	$s = \sum_{n=1}^3 \sum_{k=1}^6 \sqrt[3]{(knx^4 - ak)}$	a=1,4; x=1,5
9	$s = \sum_{n=1}^6 \sum_{k=1}^4 \operatorname{tg}^3(knx^5 - ax)$	a=1,5; x=1,6
10	$s = \sum_{n=1}^5 \sum_{k=1}^4 \sqrt[4]{(knx^3 - an)}$	a=1,1; x=1,4
11	$s = \sum_{n=1}^4 \sum_{k=1}^3 \sin^4(knx^6 - ax)$	a=1,2; x=1,8
120	$s = \sum_{n=1}^3 \sum_{k=1}^6 \operatorname{tg}^3(knx^2 - a)$	a=1,3; x=1,2
13	$s = \sum_{n=1}^5 \sum_{k=1}^4 \sqrt[4]{(knx^3 - ax)}$	a=1,4; x=1,4
14	$s = \sum_{n=1}^4 \sum_{k=1}^5 \sin^4(knx^3 - a)$	a=1,5; x=1,3
15	$s = \sum_{n=1}^3 \sum_{k=1}^7 \sqrt[3]{(knx^5 - an)}$	a=1,1; x=1,5
16	$s = \sum_{n=1}^5 \sum_{k=1}^4 \sqrt[4]{(knx^3 - ak)}$	a=1,2; x=1,9
17	$s = \sum_{n=1}^4 \sum_{k=1}^5 \sin^3(knx^6 - an)$	a=1,3; x=1,4
18	$s = \sum_{n=1}^4 \sum_{k=1}^5 \cos^4(knx^3 - ak)$	a=1,4; x=1,5

Контрольные вопросы

1. Перечислите, что должно быть в конструкции цикла.
2. С помощью каких действий реализуются циклические вычислительные процессы в Visual C++ 2010?
3. Запишите в общем виде оператор цикла **for** и опишите, как он выполняется.
4. Чем оператор цикла **for** отличается от операторов цикла **while** и **do...while**?
5. В каких случаях используется оператор цикла **for**?
6. Запишите и дайте краткую характеристику операции инкремента?
7. Сколько раз выполнится оператор цикла **int N=3; for (int i=1; i<=N; i++) N=N-1;**?
8. Сколько раз выполнится в программе оператор цикла **for (i = 0; i<1; i++) cout <<i;**?
9. Записан оператор **for (i = 2; i <10; i+=2) cout << i ;**. Что будет выведено на экран дисплея?