### Задание 7. Программирование вложенных циклов

Общие сведения

Если телом цикла является циклическая структура, то такие циклы называют *вложенными* или *сложными.*

Цикл, содержащий в себе другой цикл, называют *внешним*.

Цикл, содержащийся в теле другого цикла, называют *внутренним*.

Внутренний и внешний циклы могут быть любыми из трех рассмотренных видов: циклами с параметром, циклами с предусловием, циклами с постусловием.

Все операторы внутреннего цикла должны полностью лежать в теле внешнего цикла, циклы ни в коем случае не могут пересекаться.

Сложные циклы условно разбивают на уровни вложенности.

Внешний цикл 1 имеет уровень 0, внутренний цикл 2 – уровень 1, внутренний цикл 3 – уровень 2 и т.д.

Если в сложном цикле с глубиной вложенности *k* число повторений циклов на каждом уровне равно *N0, N1, …, Nk*соответственно, то общее количество повторений тела самого внутреннего цикла равно: *N= N0 × N1 × …× Nk*.

Примеры

***Условие задачи***: необходимо вычислить значение функции 

при следующем наборе исходных данных: *x*=5, *m*=15, *n*=10.

Код программы (Visual Studio):

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "rus");

int n,m,l=1;

double x,s,s1,s2;

cout << "введите n:"; cin >> n;

cout << "введите m:"; cin >> m;

cout << "введите x:"; cin >>x;

s = 0;

for (int j=l; j <= n; j++)

{

s1 = 0;

for (int k = l; k <= m; k++)

{

s1 = s1 + (1.\*k / j + 1.\*j / k);

}

s2 = 0;

for (int i = l; i <= m; i++)

{

s2 = s2 + (1. \* l / i);

}

s = s + (j \* x + s1) / (2 \* j + x \* s2);

}

cout << "f(x)=" << s << endl;

system("pause");

return 0;

}

***Условие задачи***: вычислить сумму бесконечного ряда



для значений *х*, изменяющихся в пределах от 0,2 до 1 с шагом 0,2. Суммирование прекращать, когда очередной член суммы по абсолютной величине станет меньше ε = 0,0001.

Задача сводится к организации вложенных циклов. Внешний цикл по счетчику обеспечивает изменение *х*. Во внутреннем цикле по условию осуществляется вычисление суммы бесконечного ряда.

Член суммы *ai* имеет достаточно сложный вид. Его целесообразно представить в виде двух сомножителей:

**, (5.1)

где  (5.2).

Здесь очередной член суммы *ci* необходимо вычислять рекуррентно. Для вывода рекуррентной формулы напишем выражения для двух последовательных членов суммы, например, *i*-го и (*i-1*)-го, и, разделив *i-й* член на (*i-1*)-й, получим выражение, на которое необходимо домножить (*i-1)*-й член для получения *i*-го. Итак,

, (5.3)

, (5.4)

 (5.5)

 (5.6)

Таким образом, чтобы получить *i*-й член из предыдущего (*i-1*)-го члена, его нужно домножить на.

Для составления программы на основе данной формулы осталось разобраться со стартовым значением *с*, то есть посчитать, чему равно *c0.* Оно должно быть вычисленопри *i=0*. В данном случае:

.

Число значений *х* на отрезке от 0,2 до 1 с шагом 0,2 равно 5. В программе для контроля при каждом значении *х* вычисляется также функция , которая приближенно может быть представлена в виде указанной суммы:

Код программы (Visual Studio):

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

double xn = 0.2, xk = 1, h = 0.2, eps = 0.0001;

double a, x, s, c;

int i;

for (x = xn; x <= xk; x = x + h)

{

s = 1; c = -1; i = 1;

do

{

c = -c \* x \* x / ((2 \* i - 1) \* 2 \* i);

a = c \* (2 \* i - 1);

s = s + a;

i = i + 1;

} while (fabs(a) >= eps);

cout << "x =" << x;

cout << "\ts =" << s << "\t";

cout<< "y =" << cos(x) + x \* sin(x) <<endl;

}

system("pause");

return 0;

Варианты заданий

**Задание 1.** В работе необходимо вычислять значение (я) функции y = f(x). Варианты заданий отличаются видом функции (табл..1). В нечетных вариантах заданий необходимо вычислять значение функции для одного значения аргумента x, а в четных следует решать задачу табулирования. При вычислении значения функции оказывается необходимым вычислять несколько сумм (произведений). Вычисление некоторых сумм (произведений) может потребовать организации вложенных циклов.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Номер | Функция | Рабочий набор | | | | |
| x | m | Xнач | xкон | n |
|  |  | 5 | 15 | - | - | 10 |
|  |  | - | 10 | 2 | 10 | 15 |
|  |  | 2 | 15 | - | - | 14 |
|  |  | - | 10 | 1 | 5 | 15 |
|  |  | 2 | 20 | - | - | 6 |
|  |  | - | 10 | 1 | 3 | 15 |
|  |  | 3 | 16 | - | - | 7 |
|  |  | - | 10 | 0 | 5 | 20 |
|  |  | 5 | 25 | - | - | 11 |
|  |  | - | 15 | 0 | 3 | 25 |
|  |  | 2 | 15 | - | - | 12 |
|  |  | - | 12 | 1 | 8 | 15 |
|  |  | 2 | 10 | - | - | 11 |
|  |  | - | 10 | 3 | 8 | 15 |
|  |  | 2 | 20 | - | - | 10 |
|  |  | - | 10 | 1 | 5 | 20 |

**Задание 2.** Вычислить сумму *s*, прекращая суммирование, когда очередной член суммы по абсолютной величине станет меньше 0,0001, при изменении аргумента *x* в указанном диапазоне [*а*, *b*] c шагом *h.* Для сравнения в каждой точке вычислить также функцию *y = f*(*x*), являющуюся аналитическим выражением ряда.

Варианты заданий

1. 
2. 
3. 
4. 
5. .
6. 
7. 
8. 
9. 
10. .
11. .
12. .
13. .
14. .
15. .
16. .
17. .

*Указание.* В задаче 2 при вычислении суммы для выхода из цикла нужно сравнивать с точностью 0,0001 не весь член суммы, а только *xi*, так как второй сомножитель при *i* = 4, 8, … равен 0, что приведет к прекращению суммирования при *i* = 4 и таким образом исказит результат.