Лабораторная работа №4.

Циклические вычислительные процессы.

Алгоритмические циклические структуры предназначены для решения задач, которые подразумевают выполнение одного и того же набора действий определённое количество раз. Выделяют три типа циклов: с параметром, с предусловием и с постусловием. Каждый цикл имеет заголовок и тело цикла.

Типы циклов

Цикл с параметром

На псевдокоде цикл записывается следующим образом:

для i=<начальное_значение> **до** <конечное_значение> **шаг** <размер_шага>
<операторы_тела_цикла>
все_для_i

Параметр цикла i показывает, сколько раз должны быть выполнены операторы тела цикла; <начальное_значение> – значение, с которого начинает изменяться параметр цикла; <конечное_значение> – значение, до которого изменяется параметр цикла; <размер_шага> – значение, показывающее, на сколько изменяется параметр цикла после выполнения всех операторов тела цикла.

Среди операторов тела цикла могут быть условные операторы, циклы и другие операторы.

Работа цикла с параметром организована по схеме: параметру присваивается <начальное_значение>, затем проверяется, больше или нет значение параметра значения <конечное_значение>. Если нет, то выполняются операторы тела цикла. В противном случае цикл завершает свою работу. После очередного выполнения операторов тела цикла значение параметра цикла изменяется на <размер_шага>. Затем опять проверяется, больше или нет значение параметра значения <конечное_значение>. Если нет, то выполняется тело цикла. В противном случае цикл завершает свою работу и т.д.

Цикл с предусловием

На псевдокоде цикл записывается следующим образом:

пока <условие>
<операторы_тела_цикла>
все_цикл

Определение <условие> аналогично его определению в разделе «**Разветвляющиеся вычислительные процессы**». Тело цикла выполняется до тех пор, пока <условие> истинно. Когда условие станет ложным, выполняется строка, следующая за циклом.

Работа цикла с предусловием:

1. Проверяется истинность выражения <условие>. Если <условие> истинно, то выполняются операторы тела цикла.

2. После того как выполнился последний оператор цикла, управление передаётся заголовку цикла. Переход на пункт 1.
3. Если условие в заголовке ложно, то цикл завершает свою работу.

Используя оператор цикла с предусловием, необходимо следить за тем, чтобы операторы тела цикла воздействовали на условие, либо за тем, чтобы оно ещё каким-то образом изменялось во время вычислений в теле цикла. Для этого часто используют унарные операции ++ или – для изменения параметров, входящих в <условие>. Только при изменении условия можно избежать зацикливания.

Цикл с постусловием

На псевдокоде цикл записывается следующим образом:

```
цикл
    <операторы_тела_цикла>
пока <условие>
```

Правила работы цикла с постусловием:

1. Выполняется тело цикла.
2. Проверяется истинность <условие>: если <условие> истинно, то выполняется тело цикла. Если оно ложно, то цикл завершает свою работу.

В противоположность циклам с параметром и предусловием, сначала проверяющим условия, в цикле с постусловием условие окончания работы этого цикла проверяется после выполнения операторов тела цикла. Данный цикл всегда выполняется, по крайней мере, один раз.

Пример. Вывести на экран таблицу из n значений функции $y = |\cos x| + |x-1|, \quad где\ x \in [a;b]$.

Алгоритм решения задачи строится следующим образом. Из условий задачи видно, что значения x изменяются от a до b. Шаг h, на который изменяются значения x, вычисляется по формуле $h = \dfrac{b-a}{n}$. Формула для изменения значений x выглядит следующим образом:

$$x = x+h$$

Следующее значение переменной   Предыдущее значение переменной

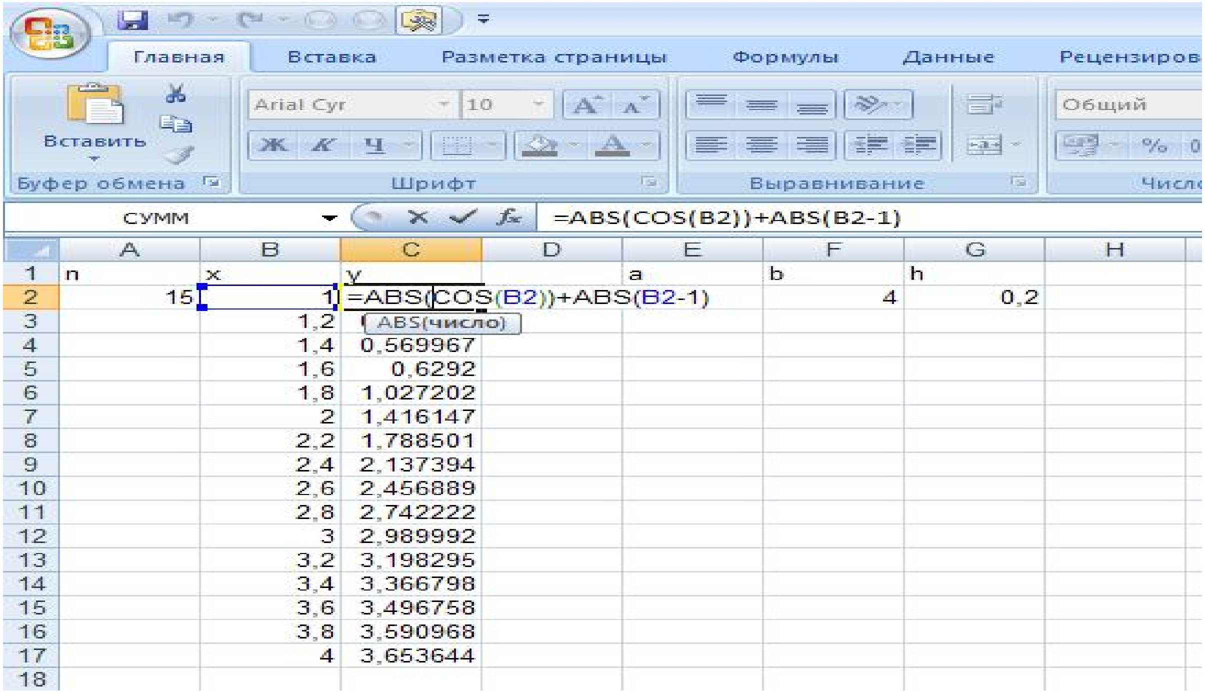
Такие формулы называют рекурсивными, поскольку новое значение переменной зависит от предыдущего значения и вычисляется через него.

Каждый раз при получении нового значения x будем вычислять значение y по заданной формуле. Таким образом, необходимо повторить одну и ту же последовательность действий известное (это важно) число раз (в данной задаче количество повторений равно n). Для решения такого типа задач как раз и используются циклические вычислительные процессы. В нашей задаче будет использован цикл с параметром.

Алгоритм

```
объявление вещ: a, b, x, y, h, цел: n, i
‘ ввод концов отрезка
ввод a
ввод b
‘ задаем количество вычисляемых значений функции
ввод n
‘ вычисление шага
h=(b-a)/n
‘ 1 вариант построения таблицы
‘ задаем начальное значение аргумента
x=a
‘ цикл для построения таблицы значений
для i=1 до n
      ‘ вычисляется значение функции
      y=|cos(x)|+|x-1|
      печать x,y
      ‘ вычисляем новое значение аргумента
      x=x+h
всё_для_i
‘ 2 вариант построения таблицы
‘ цикл для построения таблицы значений
для x=a до b  шаг h
      ‘ вычисляется значение функции
      y=|cos(x)|+|x-1|
      печать x,y
всё_для x
‘ 3 вариант построения таблицы
‘ задаем начальное значение аргумента
x=a
‘ цикл для построения таблицы значений
пока x<= b
      ‘ вычисляется значение функции
      y=|cos(x)|+|x-1|
      печать x,y
            x=x+h
всё_цикл
‘ 4 вариант построения таблицы
‘ задаем начальное значение аргумента
x=a
‘ цикл для построения таблицы значений
цикл
      ‘ вычисляется значение функции
      y=|cos(x)|+|x-1|
      печать x,y
            x=x+h
пока x<= b
```

Решение в Excel



Решение на VBA

Вариант 1

```
Dim a, b, x, y1, y2, y3, h As Single, n, i As Integer
a = InputBox("a=", , "2,5")
b = InputBox("b=", , "4,3")
n = InputBox("n=", , "10")
h = (b - a) / n
x = a
For i = 1 To n
    y1 = Abs(Cos(x))
    y2 = Abs(x - 1)
    y3 = y1 + y2
    Cells(i + 1, 1).Value = i
    Cells(i + 1, 2).Value = x
    Cells(i + 1, 3).Value = y1
    Cells(i + 1, 4).Value = y2
    Cells(i + 1, 5).Value = y3
    x = x + h
Next i
Cells(1, 1).Value = "i"
Cells(1, 2).Value = "x"
Cells(1, 3).Value = "y1"
Cells(1, 4).Value = "y2"
Cells(1, 5).Value = "y3"
```

Вариант 2

```
Dim a, b, x, y1, y2, y3, h As Single, n, i As Integer
a = CSng(InputBox("a=", , "2,5"))
b = CSng(InputBox("b=", , "4,3"))
n = CSng(InputBox("n=", , "10"))
h = (b - a) / n
x = a
i = 1
While x <= b
    y1 = Abs(Cos(x))
    y2 = Abs(x - 1)
    y3 = y1 + y2
    Cells(i + 1, 1).Value = i
    Cells(i + 1, 2).Value = x
    Cells(i + 1, 3).Value = y1
    Cells(i + 1, 4).Value = y2
    Cells(i + 1, 5).Value = y3
    x = x + h
    i = i + 1
Wend
Cells(1, 1).Value = "i"
Cells(1, 2).Value = "x"
Cells(1, 3).Value = "y1"
Cells(1, 4).Value = "y2"
Cells(1, 5).Value = "y3"
```

Вычисление последовательностей

Циклические структуры используются для вычисления элементов рекуррентных последовательностей, обработки массивов, а также решения задач, которые предполагают использование численных методов.

Определение 1. Пусть переменная x принимает последовательно значения $x_1, x_2, x_3, \dots, x_n, \dots$. Такое нумерованное множество чисел называется последовательностью. Закон образования последова-тельности задается формулой n-го члена x_n.

Из данного определения видно, что для задания последовательности нужно знать закон образования n-го члена последовательности, т.е. функцию, которая ставит натуральному числу n в соответствие некоторое значение $f(n)$: $x_n=f(n)$, *n принадлежит N*.

Определение 2. Последовательность $\{x_n\}$ называется сходящейся к x^*, если при $n\to\infty$ $|x^*-x_n|\to 0$.

Определение 3. Последовательность $\{x_n\}$ называется убывающей, если при $n\to\infty$ $|x_n|\to 0$.

Определение 4. Последовательность $\{x_n\}$ называется возрастающей, если при $n\to\infty$ $|x_n|\to\infty$.

Замечание. При вычислении возрастающих последовательностей должно быть задано условие, ограничивающее вычисление элементов такой последовательности.

Определение 5. Пусть дана некоторая последовательность элементов $a_1, a_2, a_3, \dots, a_n, \dots$, причём $a_{к+1}=F(a_1, a_2, \dots, a_к)$, $a_{к+2}=F(a_2, a_3, \dots, a_к, a_{к+1})$. Если функцию F можно определить или она задана, то последовательность $a_1, a_2, a_3, \dots, a_к, a_{к+1}, \dots$ называется рекуррентной последовательностью.

Формула n-го члена рекуррентной последовательности (рекуррентная формула) $a_n=F(a_{n-k}, a_{n-k+1}, a_{n-к+2}, \dots, a_{n-1})$, где число k называется глубиной рекурсии и определяет количество предшествующих элементов, необходимых для вычисления a_n. Смысл глубины рекурсии в том, что она максимальному количеству переменных, необходимых для вычисления элементов последовательности.

Примеры задач с использованием рекуррентных последовательностей

Вычислить n-й элемент последовательности.

Вычислить сумму или произведение первых n элементов последовательности.

Найти количество элементов на данном отрезке последовательности, удовлетворяющих определенному условию.

Найти номер первого элемента последовательности, удовлетворя-ющего определённому условию.

Пример 1. Вычислить n-й элемент арифметической прогрессии $a_1=1$, $a_2=3$, $a_3=5$ и т.д.

Написание алгоритма решения задачи будет состоять из двух шагов.

Выведем рекуррентную формулу. Так как разность прогрессии равна 2, то рекуррентная формула будет следующей: $a_i = \begin{cases} 1, & i = 1 \\ a_{i-1} + 2, & i \geq 2 \end{cases}$. Читается формула так: при $i=1$ $a_i=1$; при $i\geq 2$ $a_i=a_{i-1}+2$, т.е. каждый последующий элемент равен сумме предыдущего и 2.

Определим глубину рекурсии. Поскольку каждый следующий элемент вычисляется только через один предыдущий, то глубина рекурсии равна 1. Следовательно, для вычисления элементов последовательности нужны две переменные. Однако в этом случае можно обойтись одной переменной. Для определения значения последующего элемента последовательности будем использовать рекурсивную формулу

$$a = a + 2$$

Следующее значение переменной

Предыдущее значение переменной

Алгоритм

```
объявление цел: a, n, i
ввод n
‘текущий элемент последовательности и его  текущий номер
a=1, i=1
‘вычисляем элементы последовательности
для i=2 до n шаг 1
      a=a+2
всё_для i
печать a
```

Пример 2. Вывести на печать первые n $(n \geq 3)$ чисел Фибоначчи. Распечатать все элементы и подсчитать, сколько среди них четных чисел. Числа Фибоначчи образуются по закону:

$$a_1=1,\ a_2=1,\ \ldots,\ a_i=a_{i-1}+a_{i-2},\ \ldots.$$

Написание алгоритма решения задачи будет состоять из двух шагов.

Рекуррентная формула задается в определении чисел Фибоначчи:

$$a_i = \begin{cases} 1, & i=1 \\ 1, & i=2 \\ a_{i-1}+a_{i-2}, & i \geq 3 \end{cases}.$$

Глубина рекурсии равна 2, поэтому для вычисления чисел Фибоначчи нужны три переменные.

Алгоритм

```
объявление цел: i, n, a1, a2, a3, s
‘ввод количества вычисляемых элементов
ввод n
‘инициализация
a1=1, a2=1, s=0
печать a1
печать a2
‘цикл для вычисления элементов и количества ‘четных элементов
для i=3 до n  шаг 1
   a3= a2+ a1
   печать a3
   если (остаток от деления a3 на 2)=0 ‘условие четности значения переменной a3
      s=s+1
   все_если
    a1= a2
   a2 = a3
всё_для i
печать s
```

Написать программу, соответствующую алгоритму

Примечание. В алгоритме использована рекурсивная формула

$$s = s + 1$$

Следующее значение переменной   Предыдущее значение переменной

Замечание. Последовательности в примерах 1 и 2 являются возрастающими, поэтому вычисление элементов ограничивается заданием их количества.

Пример 3. Для заданных действительного x и целого n вычислить сумму $s = \frac{\sin x}{x} - \frac{\sin^2 x}{x^2} + \frac{\sin^3 x}{x^3} - \ldots + (-1)^{n-1} \frac{\sin^n x}{x^n}$.

Написание алгоритма решения задачи будет состоять из двух шагов.

Формула для вычисления суммы:

$$a_i = \begin{cases} \frac{\sin x}{x}, & i = 1 \\ (-1)\frac{\sin x}{x} a_{i-1}, & i \geq 2 \end{cases}.$$

Здесь i обозначает номер текущего элемента последовательности, n – количество элементов последовательности, сумму которых нужно вычислить.

Глубина рекурсии в данном случае не определяется, так как данная формула не является рекуррентной.

Алгоритм

```
Объявление цел: i, n; вещ: x, s, a
‘ввод количества элементов
ввод n
ввод x
‘начальное значение номера элемента
i=1
‘начальное значение элемента
a=sin(x)/x
‘начальное значение суммы элементов
s=a
‘цикл для вычисления элементов и их суммы
для i=2 до n  шаг 1
   a= (-1)sin(x)/x*a
   s=s+a
всё_для i
печать s
```

Написать программу, соответствующую алгоритму

```
Dim i, n As Integer, x, s, a As Single
n = CInt(InputBox("n="))
x = CSng(InputBox("x="))
i = 1
a = Sin(x) / x
s = a
For i = 2 To n
   a = (-1) * Sin(x)/x * a
   s = s + a
Next i
MsgBox "s=" & CStr(s)
```

Примечание. В алгоритме использована рекурсивная формула

s= s + a

Следующее значение переменной   Предыдущее значение переменной

Задание. Разработать алгоритм и по нему составить программу для вычисления заданной величины, определив её зависимость от параметра. Из условий задачи найти начальное и конечное значения параметра, а также шаг изменения параметра.

1. Дано натуральное число *N*. Вычислить $S = 1 - \frac{1}{2} + \frac{1}{4} - \frac{1}{8} + ... + (-1)^N \cdot \frac{1}{2^N}$
2. Дано натуральное число *N*. Вычислить

   $$S = \frac{1}{\sin 1} + \frac{1}{\sin 1 + \sin 2} + ... + \frac{1}{\sin 1 + \sin 2 + ... + \sin N}$$

3. Дано натуральное число *N*. Вычислить произведение $P = \frac{2}{3} \cdot \frac{4}{5} \cdot \frac{6}{7} \cdot ... \cdot \frac{2N}{2N+1}$
4. Дано натуральное число *N*. Вычислить

$$P = \frac{\cos 1}{\sin 1} \cdot \frac{\cos 1 + \cos 2}{\sin 1 + \sin 2} \cdot \ldots \cdot \frac{\cos 1 + \cos 2 + \ldots + \cos N}{\sin 1 + \sin 2 + \ldots + \sin N}$$

5. Дано действительное число x и натуральное число N. Вычислить

$$S = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots + (-1)^{2N+1}\frac{x^{2N+1}}{(2N+1)!}$$

6. Даны натуральное число n и действительное число x. Вычислить

$$S = \sin x + \sin\sin x + \ldots + \underbrace{\sin\sin\ldots\sin x}_{n\ раз}$$

7. Даны действительное число a и натуральное число n. Вычислить

$P=a(a+1)\ldots(a+n-1)$

8. Даны действительное число a и натуральное число n. Вычислить

$P=a(a-n)(a-2n)\ldots(a-n^2)$

9. Даны действительное число а и натуральное число n. Вычислить

$$S = \frac{1}{a} + \frac{1}{a^2} + \frac{1}{a^4} + \ldots + \frac{1}{a^{2n-2}}$$

10. Дано действительное число x. Вычислить $Q = \dfrac{(x-1)(x-3)(x-7)\ldots(x-63)}{(x-2)(x-4)(x-8)\ldots(x-64)}$

11. Вычислить $P=(1+\sin 0.1)(1+\sin 0.2)\ldots(1+\sin 10)$

12. Даны натуральное число n и действительное число x. Вычислить

$S=\sin x+\sin x^2+\ldots+\sin x^n$

13. Дано натуральное число n. Вычислить $S=1\cdot 2+2\cdot 3\cdot 4+\ldots+n\cdot(n+1)\cdot\ldots\cdot 2n$

14. Дано натуральное число $n>2$. Вычислить $P = (1-\frac{1}{2^2})(1-\frac{1}{3^2})\cdot\ldots\cdot(1-\frac{1}{n^2})$

15. Дано натуральное число n. Вычислить $P = (1-\frac{1}{2})(1-\frac{1}{4})(1-\frac{1}{6})\ldots(1-\frac{1}{2n})$

16. Дано натуральное число $n>1$. Вычислить $S=1!+2!+3!+\ldots+n!$

17. Дано натуральное число n. Вычислить $S = \frac{1}{3^2} + \frac{1}{5^2} + \frac{1}{7^2} + \ldots + \frac{1}{(2n+1)^2}$

18. Для данного действительного числа x вычислить по схеме Горнера

$y=x^{10}+2x^9+3x^8+\ldots+10x+11$

19. Числа Фибоначчи (f_n) определяются формулами $f_0=f_1=1, f_n=f_{n-1}+f_{n-2}$,

$n=2, 3, \ldots$ Определить f_{40}.

20. Дано натуральное число *n*. Вычислить $Y=1\cdot2\cdot3\cdot\ldots\cdot(2n-1)$

21. Дано натуральное число *n*. Вычислить $Y=2\cdot4\cdot6\cdot\ldots\cdot(2n)$

22. Вычислить $y=\cos x+\cos x^2+\cos x^3+\ldots+\cos x^n$

23. Вычислить $y=\sin1+\sin1.1+\sin1.2+\ldots+\sin2$

24. Дано натуральное число *n*. Вычислить $S=\frac{2}{1}+\frac{3}{2}+\frac{4}{3}+\ldots+\frac{n+1}{n}$