# Министерство образования Республики Беларусь

Учреждение образования

БелорусскиЙ государственный университет

информатики и радиоэлектроники

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

|  |
| --- |
|  |
|  |

# оТЧЕТ

по лабораторной работе

на тему:

Сложные циклы

|  |  |  |
| --- | --- | --- |
| Выполнил  Студент гр. 451001 |  | В. А. Пузик |
| Проверил |  | Асс. Е.Е. Фадеева |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Минск, 2018

1. Теоретические сведения по теме лабораторной работы

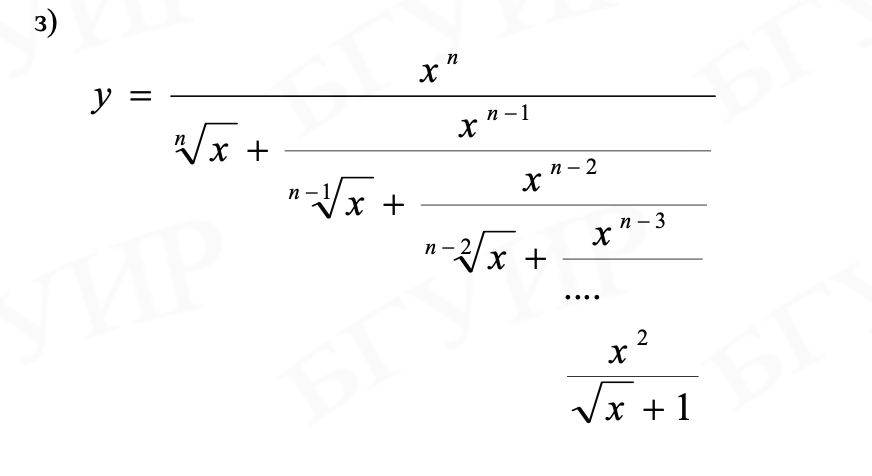
Типы циклов:

1. По типу вложенности
   1. Внешние циклы — циклы, не имеющие родительских, не вложенные в другой цикл
   2. Вложенный цикл — циклы, вложенные в другой цикл
   3. сложный цикл — конструкция, состоящая из внешнего и как минимум одного ложенного цикла
2. По типу счетчика
   1. цикл со счетчиком — цикл, выполняющийся фиксированное кол-во раз
   2. итерационный цикл — цикл, выполняющийся до тех пор, пока выполняется определенное условие, и количество итераций может быть неопределенным заранее.

Алгоритм - система правил, четко описывающая последовательность действий, которые необходимо выполнить для решения задачи.

Свойства правильного алгоритма:

1. Дискретность - любой алгоритм состоит из отдельных операций (этапов, действий), которые выполняются дискретно (по шагам).
2. Определенность — свойство алгоритма, которое гласит, что каждый его шаг должен быть строго определен и различные толкования должны быть исключены.
3. Результативность — свойство, которое гласит, что любой алгоритм должен завершаться с результатом.
4. Массовость — свойство, которое гласит, что любой алгоритм можно успешно применять к различным наборам исходных данных в определенном формате.
5. Задание на лабораторную работу
   1. Постановка задачи

Вычислить цепную дробь для X=0.5(0.05)0.8  


* 1. Эскиз ожидаемого результата

N: 5

n=5 x=0.50 y=0.03

n=5 x=0.55 y=0.05

n=5 x=0.60 y=0.08

n=5 x=0.65 y=0.11

n=5 x=0.70 y=0.15

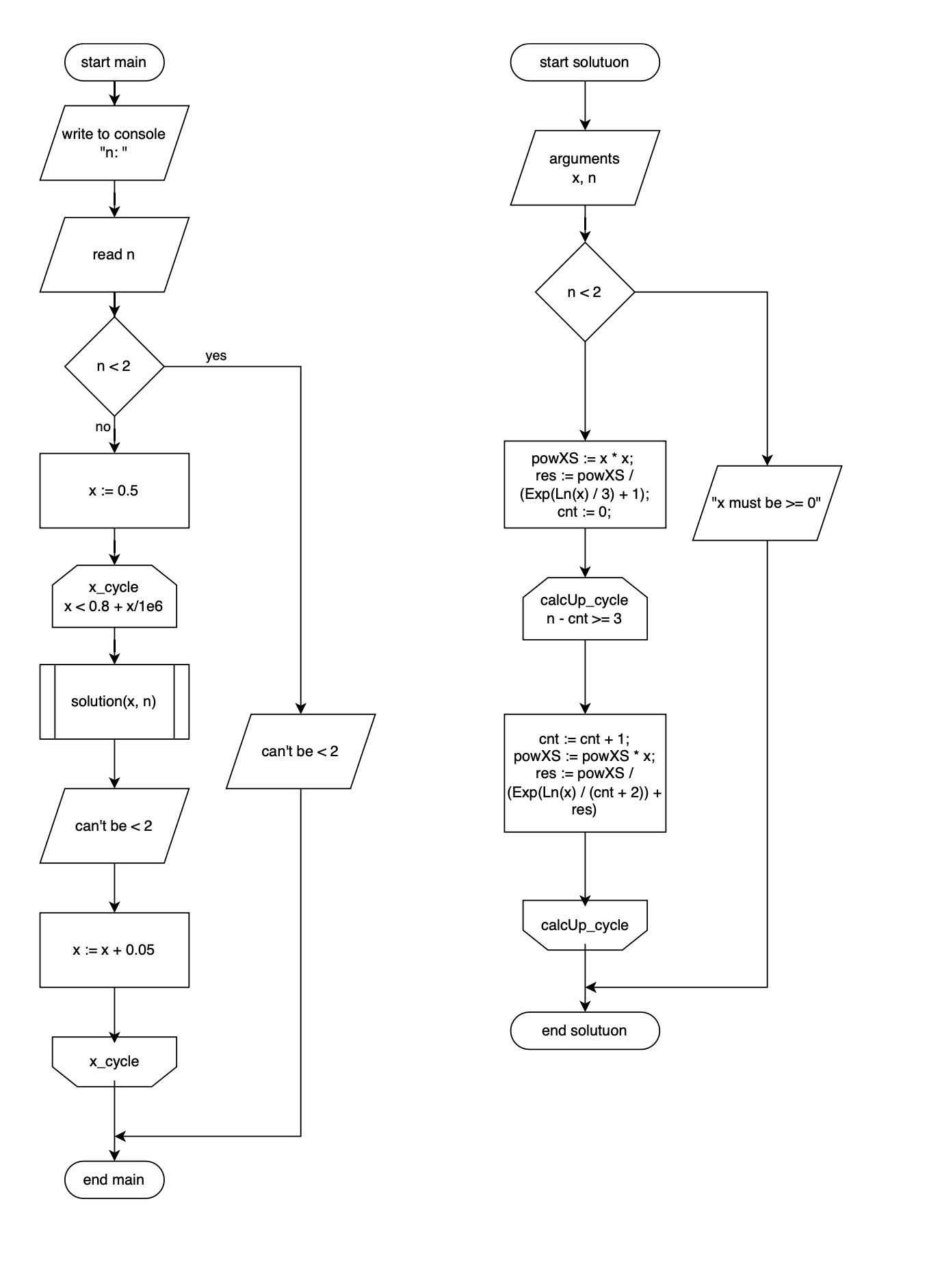
n=5 x=0.75 y=0.20

n=5 x=0.80 y=0.26

1. Выполнение
   1. Разработка алгоритма

Таблица 3.1 используемые идентификаторы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя идентификатора | назначение | Тип идентификатора | Начальное значение | Закон изменения | Имя цикла, в котором происходит изменение переменной |
| n | Параметр дроби | int | - | Read(n) | - |
| x | Параметр дроби | int | 0.5 | x := x + 0.05; | x\_cycle |
| powXS | Текущее значение посчитанной степени | Double | x \* x | powXS := powXS \* x; | calcUp\_cycle |
| res | Промежуточное значение результата | Double | powXS / (Exp(Ln(x) / 3) + 1) | res := powXS / (Exp(Ln(x) / (cnt + 2)) + res) | calcUp\_cycle |
| cnt | Кол-во проходок цикла | Integer | 0 | cnt := cnt + 1 | calcUp\_cycle |



Для цикла MainCycle был выбран оператор while, так как заранее неизвестно сколько раз будет необходимо сгенерировать решение перед тем, как оно пройдет проверку. Цикл LengthCycle и PositionCycle и GenCycle реализованы при помощи цикла for так как сразу известно сколько итераций займет проход по каждому из вышеописанных циклов.

* 1. Текст программы и его описание

program SolveExpression;

uses

SysUtils;

function solution(x: Double; n: Integer): Double;

var

powXS: Double;

res: Double;

cnt: Integer;

begin

if (x < 0) then

begin

WriteLn('x must be >= 0')

end

else

begin

powXS := x \* x;

res := powXS / (Exp(Ln(x) / 3) + 1);

cnt := 0;

while (n - cnt >= 3) do

begin

cnt := cnt + 1;

powXS := powXS \* x;

res := powXS / (Exp(Ln(x) / (cnt + 2)) + res)

end

end;

Result := res;

end;

procedure main();

var

n: Integer;

x: Double;

begin

WriteLn('n: ');

Read(n);

if (n < 2) then

begin

WriteLn('cant be < 2')

end

else

begin

x := 0.5;

while(x < 0.8 + x/1e6) do

begin

WriteLn(Format('n=%d x=%f y=%f', [n, x, solution(x, n)]));

x := x + 0.05;

end

end;

ReadLn;

ReadLn

end;

begin

main;

end.

* 1. Тестирование и отладка программы

Таблица 3.2 Прохождение тестов программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Специфика тестирования | Номер теста | Вводимые данные | Ожидаемый результат | Полученный результат |
| Корректность работы программы | 1 | N: 10 | n: 4  n=4 x=0.50 y=0.06  n=4 x=0.55 y=0.09  n=4 x=0.60 y=0.12  n=4 x=0.65 y=0.16  n=4 x=0.70 y=0.20  n=4 x=0.75 y=0.25  n=4 x=0.80 y=0.30 | n: 4  n=4 x=0.50 y=0.06  n=4 x=0.55 y=0.09  n=4 x=0.60 y=0.12  n=4 x=0.65 y=0.16  n=4 x=0.70 y=0.20  n=4 x=0.75 y=0.25  n=4 x=0.80 y=0.30 |

* 1. Итоговый текст программы

program SolveExpression;

{$APPTYPE CONSOLE}

// y(x, n) = pow(x, n) / (root(x, n) + (pow(x, n-1) / (root(x, n-2) + pow(x, n-3) / (root(x, n-3) + pow(x, n-4)))) .... / (root(x, 2) + pow(x, 1))

// y(x, 2) = pow(x, 2) /

// (root(x, 2) + pow(x, 1) /

// (root(x, 1) + pow(x, 0))

uses

SysUtils;

function solution(x: Double; n: Integer): Double;

var

powXS: Double;

res: Double;

cnt: Integer;

begin

if (x < 0) then

begin

WriteLn('x must be >= 0')

end

else

begin

powXS := x \* x;

res := powXS / (Exp(Ln(x) / 3) + 1);

cnt := 0;

while (n - cnt >= 3) do

begin

cnt := cnt + 1;

powXS := powXS \* x;

res := powXS / (Exp(Ln(x) / (cnt + 2)) + res)

end

end;

solution := res;

end;

procedure main();

var

n: Integer;

x: Double;

begin

WriteLn('n: ');

Read(n);

if (n < 2) then

begin

WriteLn('cant be < 2')

end

else

begin

x := 0.5;

while(x < 0.8 + x/1e6) do

begin

WriteLn(Format('n=%d x=%f y=%f', [n, x, solution(x, n)]));

x := x + 0.05;

end

end;

ReadLn;

ReadLn

end;

begin

main;

end.