# 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:

Итерационные процессы

|  |  |  |
| --- | --- | --- |
| 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 значения функций *f*1*(x)* и *f*2*(x)* для значений аргументов, указанных в пунктах заданий №1-30. Функцию *f*2*(x)* вычислить для ряда точностей . Для указанных точностей определить количество N элементов ряда, суммируемых для достижения заданной точности. Результаты расчетов свести в следующую таблицу:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *x* | *f*1*(x)* |  | |  | |  | |
| *f*2*(x)* | N | *f*2*(x)* | N | *f*2*(x)* | N |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| …. |  |  |  |  |  |  |  |

Для аргумента , изменяющегося от 0 с шагом 0.04 вычислить 20 значений функций:





* 1. Эскиз ожидаемого результата

┌──────┬────────┬─────────────┬─────────────┬─────────────┐

│ x │ f1(x) │ e=0.01 │ e=0.001 │ e=0.0001 │

│ │ ├────────┬────┼────────┬────┼────────┬────┤

│ │ │ f2(x) │ N │ f2(x) │ N │ f2(x) │ N │

├──────┼────────┼────────┼────┼────────┼────┼────────┼────┤

│ 0.00 │ 0.6931 │ 0.6931 │ 2 │ 0.6931 │ 2 │ 0.6931 │ 2 │

├──────┼────────┼────────┼────┼────────┼────┼────────┼────┤

│ 0.04 │ 0.6935 │ 0.6935 │ 2 │ 0.6935 │ 2 │ 0.6935 │ 2 │

├──────┼────────┼────────┼────┼────────┼────┼────────┼────┤

│ 0.08 │ 0.6947 │ 0.6947 │ 2 │ 0.6947 │ 2 │ 0.6947 │ 2 │

├──────┼────────┼────────┼────┼────────┼────┼────────┼────┤

│ 0.12 │ 0.6967 │ 0.6967 │ 2 │ 0.6967 │ 2 │ 0.6967 │ 2 │

├──────┼────────┼────────┼────┼────────┼────┼────────┼────┤

│ 0.16 │ 0.6995 │ 0.6994 │ 2 │ 0.6994 │ 2 │ 0.6994 │ 3 │

├──────┼────────┼────────┼────┼────────┼────┼────────┼────┤

│ 0.20 │ 0.7030 │ 0.7028 │ 2 │ 0.7028 │ 2 │ 0.7029 │ 3 │

├──────┼────────┼────────┼────┼────────┼────┼────────┼────┤

│ 0.24 │ 0.7072 │ 0.7069 │ 2 │ 0.7069 │ 2 │ 0.7070 │ 3 │

├──────┼────────┼────────┼────┼────────┼────┼────────┼────┤

│ 0.28 │ 0.7122 │ 0.7116 │ 2 │ 0.7117 │ 3 │ 0.7117 │ 3 │

├──────┼────────┼────────┼────┼────────┼────┼────────┼────┤

│ 0.32 │ 0.7178 │ 0.7168 │ 2 │ 0.7169 │ 3 │ 0.7169 │ 4 │

├──────┼────────┼────────┼────┼────────┼────┼────────┼────┤

│ 0.36 │ 0.7241 │ 0.7224 │ 2 │ 0.7227 │ 3 │ 0.7227 │ 4 │

├──────┼────────┼────────┼────┼────────┼────┼────────┼────┤

│ 0.40 │ 0.7309 │ 0.7283 │ 2 │ 0.7290 │ 3 │ 0.7289 │ 4 │

├──────┼────────┼────────┼────┼────────┼────┼────────┼────┤

│ 0.44 │ 0.7384 │ 0.7345 │ 2 │ 0.7355 │ 4 │ 0.7355 │ 5 │

├──────┼────────┼────────┼────┼────────┼────┼────────┼────┤

│ 0.48 │ 0.7463 │ 0.7408 │ 2 │ 0.7423 │ 4 │ 0.7424 │ 5 │

├──────┼────────┼────────┼────┼────────┼────┼────────┼────┤

│ 0.52 │ 0.7548 │ 0.7501 │ 3 │ 0.7494 │ 4 │ 0.7495 │ 6 │

├──────┼────────┼────────┼────┼────────┼────┼────────┼────┤

│ 0.56 │ 0.7637 │ 0.7579 │ 3 │ 0.7570 │ 5 │ 0.7569 │ 7 │

├──────┼────────┼────────┼────┼────────┼────┼────────┼────┤

│ 0.60 │ 0.7730 │ 0.7661 │ 3 │ 0.7646 │ 5 │ 0.7644 │ 7 │

├──────┼────────┼────────┼────┼────────┼────┼────────┼────┤

│ 0.64 │ 0.7827 │ 0.7710 │ 4 │ 0.7719 │ 6 │ 0.7720 │ 8 │

├──────┼────────┼────────┼────┼────────┼────┼────────┼────┤

│ 0.68 │ 0.7927 │ 0.7779 │ 4 │ 0.7798 │ 7 │ 0.7797 │ 9 │

├──────┼────────┼────────┼────┼────────┼────┼────────┼────┤

│ 0.72 │ 0.8030 │ 0.7843 │ 4 │ 0.7872 │ 8 │ 0.7874 │ 11 │

├──────┼────────┼────────┼────┼────────┼────┼────────┼────┤

│ 0.76 │ 0.8136 │ 0.7978 │ 5 │ 0.7953 │ 9 │ 0.7951 │ 13 │

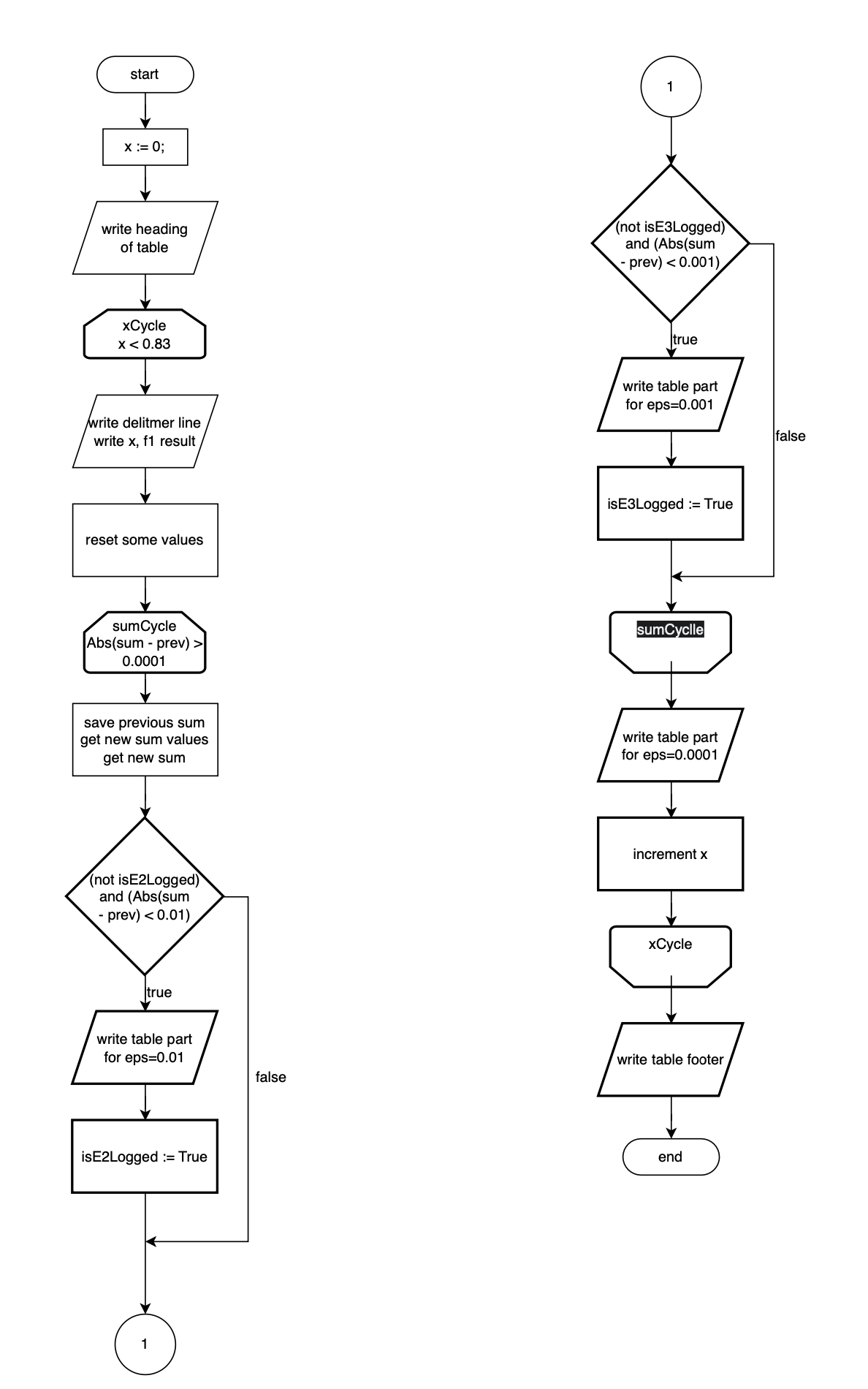
├──────┼────────┼────────┼────┼────────┼────┼────────┼────┤

│ 0.80 │ 0.8244 │ 0.7998 │ 6 │ 0.8029 │ 11 │ 0.8027 │ 15 │

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 |
| k | Кол-во шагов подсчета суммы | int | 1 | 1) k := 1;  2) Inc(k) | 1) xCycle  2) sumCycle |
| x | Параметр дроби | double | -1 | x := x + 0.04 | xCycle |
| curx2k | Текущее значение x^(2k) | double | x \* x | 1) curx2k := x \* x;  2) curx2k := curx2k \* x \* x; | 1) xCycle  2) sumCycle |
| prev | Предыдущее значение суммы | double | 1.7e308 | 1) prev := 1.7e308;  2) prev := sum; | sumCycle |
| cur1 | Текущее значение (-1)^k | double | -1 | 1) cur1 := -cur1;  2) cur1 := -1; | 1) xCycle  2) sumCycle |
| cur22k | Текущее значение 2^(2k) | double | 4 | 1) cur22k := 4;  2) cur22k := cur22k \* 2 \* 2; | 1) xCycle  2) sumCycle |
| curFact | Текущее значение k! | double | 1 | 1) curFact := 1;  2) curFact := curFact \* k; | 1) xCycle  2) sumCycle |
| cur2k1Fact | Текущее значение (2k-1)! | double | 1 | 1) cur2k1Fact := 1;  2) cur2k1Fact := cur2k1Fact \* (2 \* k) \* (2 \* k - 1); | 1) xCycle  2) sumCycle |
| sum | Текущее значение суммы | double | (cur1 \* cur2k1Fact \* curx2k) / (cur22k \* curFact \* curFact) | 1) sum := (cur1 \* cur2k1Fact \* curx2k) / (cur22k \* curFact \* curFact);  2) sum := sum + (cur1 \* cur2k1Fact \* curx2k) / (cur22k \* curFact \* curFact); | 1) xCycle  2) sumCycle |
| isE2Logged | Выводилась ли часть таблицы для eps=0.01 | Boolean | False | 1) isE2Logged := False;  2) isE2Logged := True; | 1) xCycle  2) sumCycle |
| isE3Logged | Выводилась ли часть таблицы для eps=0.001 | Boolean | False | 1) isE3Logged := False;  2) isE2Logged := True; | 1) xCycle  2) sumCycle |



* 1. Текст программы и его описание

uses

SysUtils;

var

x, sum, prev, cur1, cur22k, curFact, cur2k1Fact, curx2k: Double;

k: Integer;

isE2Logged, isE3Logged: Boolean;

begin

x := 0;

WriteLn('┌──────┬────────┬─────────────┬─────────────┬─────────────┐');

WriteLn('│ x │ f1(x) │ e=0.01 │ e=0.001 │ e=0.0001 │');

WriteLn('│ │ ├────────┬────┼────────┬────┼────────┬────┤');

WriteLn('│ │ │ f2(x) │ N │ f2(x) │ N │ f2(x) │ N │');

{ xCycle }

while x < 0.83 do

begin

WriteLn('├──────┼────────┼────────┼────┼────────┼────┼────────┼────┤');

Write(Format('│ %.2f │ %.4f', [x, Ln(1 + Sqrt(1 + x \* x))]));

k := 1;

cur1 := -1;

cur22k := 4;

curFact := 1;

cur2k1Fact := 1;

curx2k := x \* x;

sum := (cur1 \* cur2k1Fact \* curx2k) / (cur22k \* curFact \* curFact);

prev := 1.7e308;

isE2Logged := False;

isE3Logged := False;

{ sumCycle }

while Abs(sum - prev) > 0.0001 do

begin

prev := sum;

Inc(k);

cur1 := -cur1;

cur22k := cur22k \* 2 \* 2;

curFact := curFact \* k;

cur2k1Fact := cur2k1Fact \* (2 \* k) \* (2 \* k - 1);

curx2k := curx2k \* x \* x;

sum := sum + (cur1 \* cur2k1Fact \* curx2k) / (cur22k \* curFact \* curFact);

if (not isE2Logged) and (Abs(sum - prev) < 0.01) then

begin

Write(Format(' │ %.4f │ %2d', [Ln(2) - sum, k]));

isE2Logged := True;

end;

if (not isE3Logged) and (Abs(sum - prev) < 0.001) then

begin

Write(Format(' │ %.4f │ %2d', [Ln(2) - sum, k]));

isE3Logged := True;

end;

end;

Write(Format(' │ %.4f │ %2d │', [Ln(2) - sum, k]));

WriteLn();

x := x + 0.04;

end;

WriteLn('└──────┴────────┴────────┴────┴────────┴────┴────────┴────┘');

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 | - |  |  |

* 1. Итоговый текст программы

uses

SysUtils;

var

x, sum, prev, cur1, cur22k, curFact, cur2k1Fact, curx2k: Double;

k: Integer;

isE2Logged, isE3Logged: Boolean;

begin

x := 0;

WriteLn('┌──────┬────────┬─────────────┬─────────────┬─────────────┐');

WriteLn('│ x │ f1(x) │ e=0.01 │ e=0.001 │ e=0.0001 │');

WriteLn('│ │ ├────────┬────┼────────┬────┼────────┬────┤');

WriteLn('│ │ │ f2(x) │ N │ f2(x) │ N │ f2(x) │ N │');

{ xCycle }

while x < 0.83 do

begin

WriteLn('├──────┼────────┼────────┼────┼────────┼────┼────────┼────┤');

Write(Format('│ %.2f │ %.4f', [x, Ln(1 + Sqrt(1 + x \* x))]));

k := 1;

cur1 := -1;

cur22k := 4;

curFact := 1;

cur2k1Fact := 1;

curx2k := x \* x;

sum := (cur1 \* cur2k1Fact \* curx2k) / (cur22k \* curFact \* curFact);

prev := 1.7e308;

isE2Logged := False;

isE3Logged := False;

{ sumCycle }

while Abs(sum - prev) > 0.0001 do

begin

prev := sum;

Inc(k);

cur1 := -cur1;

cur22k := cur22k \* 2 \* 2;

curFact := curFact \* k;

cur2k1Fact := cur2k1Fact \* (2 \* k) \* (2 \* k - 1);

curx2k := curx2k \* x \* x;

sum := sum + (cur1 \* cur2k1Fact \* curx2k) / (cur22k \* curFact \* curFact);

if (not isE2Logged) and (Abs(sum - prev) < 0.01) then

begin

Write(Format(' │ %.4f │ %2d', [Ln(2) - sum, k]));

isE2Logged := True;

end;

if (not isE3Logged) and (Abs(sum - prev) < 0.001) then

begin

Write(Format(' │ %.4f │ %2d', [Ln(2) - sum, k]));

isE3Logged := True;

end;

end;

Write(Format(' │ %.4f │ %2d │', [Ln(2) - sum, k]));

WriteLn();

x := x + 0.04;

end;

WriteLn('└──────┴────────┴────────┴────┴────────┴────┴────────┴────┘');

end.