Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Основы алгоритмизации и программирования (ОАиП)

ОТЧЕТ

по лабораторной работе №5

Тема работы:

Вычисление бесконечного ряда

Выполнил

студент: гр. 151004 Данилов Ф.А.

Проверил: Фадеева Е.П.

Минск 2021

Содержание

[1 Постановка задачи 3](#_Toc89547600)

[2 Методика решения 4](#_Toc89547601)

[2.1 Доказательство сходимости ряда 4](#_Toc89547602)

[2.2 Алгоритм 4](#_Toc89547603)

[3 Текстовый алгоритм решения задачи 5](#_Toc89547604)

[4 Структура данных 6](#_Toc89547605)

[5 Схема алгоритма решения задачи по ГОСТ 19.701-90 7](#_Toc89547606)

[6 Результаты расчетов 11](#_Toc89547607)

[Приложение А 12](#_Toc89547608)

[Приложение Б 15](#_Toc89547609)

# Постановка задачи

Для заданной функции бесконечного ряда:

вычислить её значение при значении , изменяющемся от до с шагом и точностях и .

Вывести на печать результаты расчётов:

значение значение при значение при

# Методика решения

## Доказательство сходимости ряда

Перед началом рассчетов суммы ряда с определенной точностью необходимо проверить что ряд сходим. Для этого используется признак сходимости Даламбера который гласит, что если существует предел отношения последующего члена ряда к предыдущему , то:

* при  ряд **сходится**. В частности, ряд сходится при ;
* при  ряд **расходится**. В частности, ряд расходится при ;
* при  **признак ответа не дает.**

**Доказательство:**

Так как для всех в промежутке с шагом , то ряд сходится.

## Алгоритм

Для расчета значений ряда используется цикл Repeat…Until с постусловием с вложенным циклом Repeat…Until с постусловием. Проверка точности вычислений происходит путем вычитания предыдущего значения суммы ряда из текущего.

# Текстовый алгоритм решения задачи

Таблица – Алгоритм решения

|  |  |
| --- | --- |
| Номер  шага | Назначение шага |
|  | EPS1 = 0.00001 |
|  | EPS2 = 0.000001 |
|  | XS := 0.1 |
|  | XF := 0.9 |
|  | XH := 0.1 |
|  | X := XS |
|  | Начало цикла А1. |
|  | Вывод X |
|  | Y := 0 |
|  | Denum := 1 |
|  | k := 0 |
|  | Frac := X |
|  | HasReached := false |
|  | Начало цикла A2 |
|  | y := y + Frac |
|  | k := k + 1 |
|  | Num := exp((4\*k+1)\*ln(X)) |
|  | i := 0 |
|  | Начало цикла А3. Проверка выполнения условия (i ≤ 3). Если условие истинно, идти к шагу 20, иначе – к шагу 23 |
|  | Denum := Denum \* (4\*k+1 - i) |
|  | i := i + 1 |
|  | Конец цикла А3 |
|  | Frac := Num / Denum |
|  | Проверка выполнения условия (not HasReached and Frac ≤ Eps1). Если условие истинно, идти к шагу 25, иначе – к шагу 27 |
|  | HasReached := true |
|  | Вывод Y |
|  | Конец цикла А2. Проверка выполнения условия (Frac ≤ Eps2). Если условие истинно, идти к шагу 28, иначе – к шагу 14 |
|  | Вывод Y |
|  | X := X + HX |
|  | Конец цикла А1. Проверка выполнения условия (X > XF). Если условие истинно, идти к шагу 31, иначе – к шагу 7 |
|  | Останов. |

# Структура данных

Таблица – Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| Eps1 | Real | Значение первой точности |
| Eps2 | Real | Значение второй точности |
| Y | Real | Функция |
| X | Real | Аргумент функции |
| XS | Real | Начальное значение |
| XF | Real | Конечное значение |
| HX | Real | Приращение |
| Num | Real | Числитель |
| Denum | Integer | Знаменатель |
| Frac | Real | Элемент суммы |
| k | Integer | Счетчик суммы |
| i | Integer | Счетчик подсчета факториала |
| HasReached | Boolean | Логическая переменная для проверки достижения |

# Схема алгоритма решения задачи по ГОСТ 19.701-90

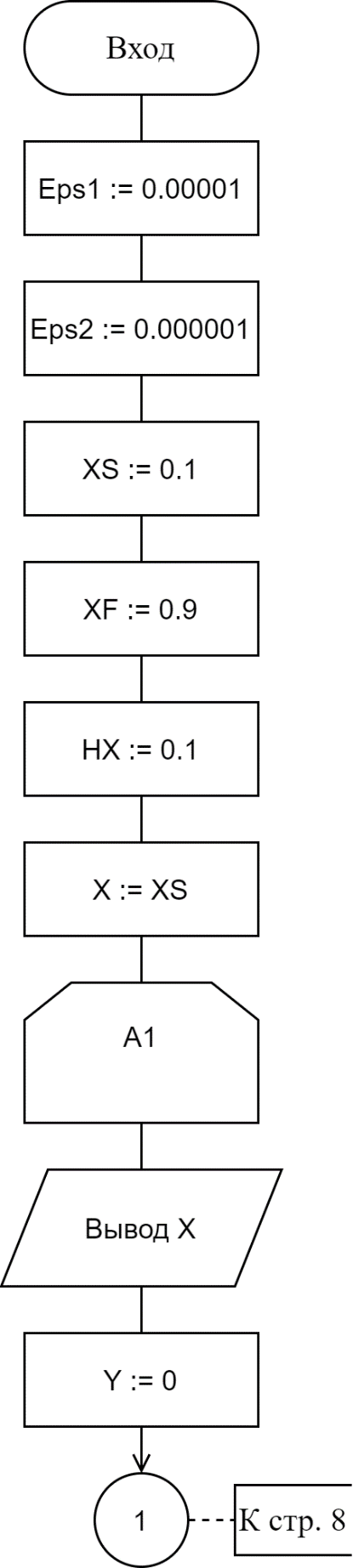


Рисунок - Схема алгоритма решения задачи по ГОСТ 19.701-90 (часть 1)

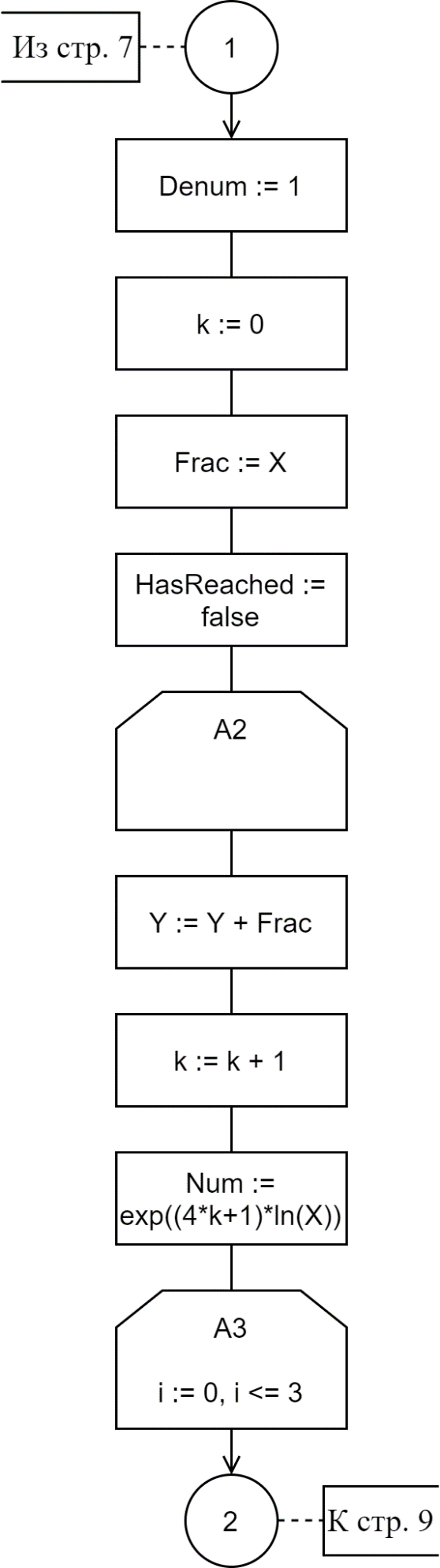


Рисунок - Схема алгоритма решения задачи по ГОСТ 19.701-90 (часть 2)

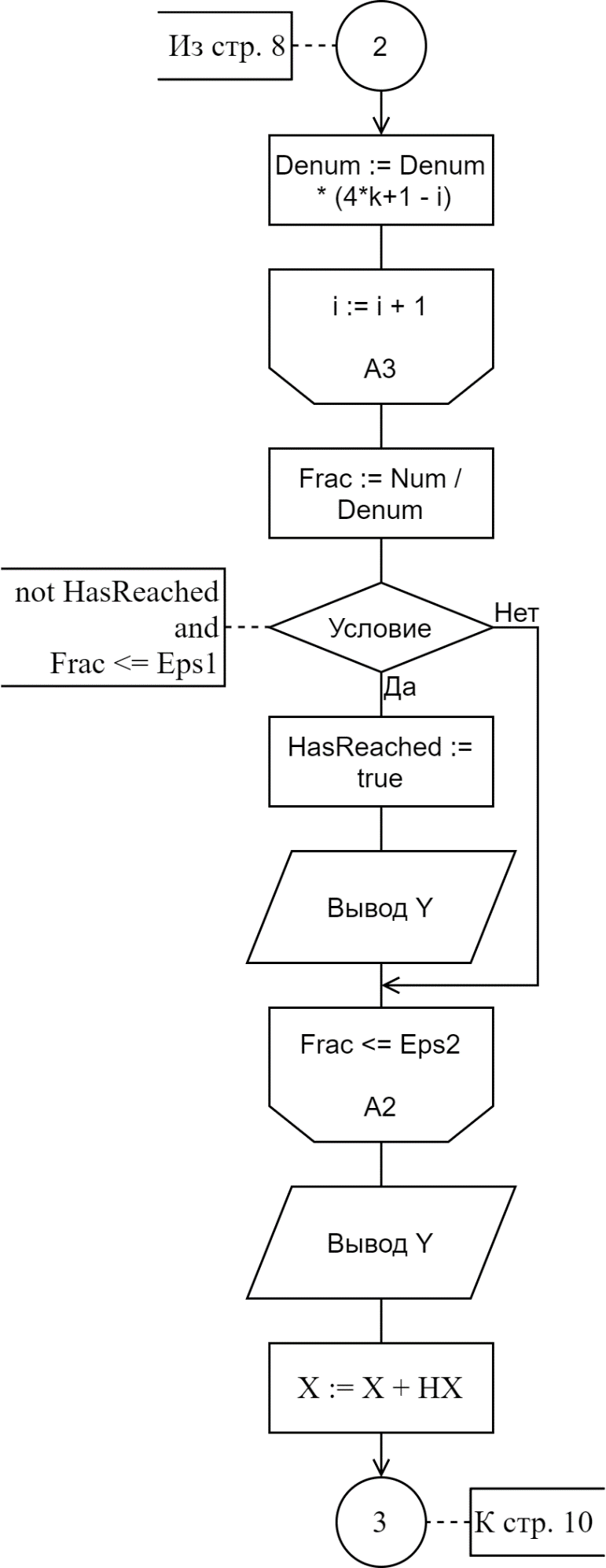


Рисунок 3 - Схема алгоритма решения задачи по ГОСТ 19.701-90 (часть 3)

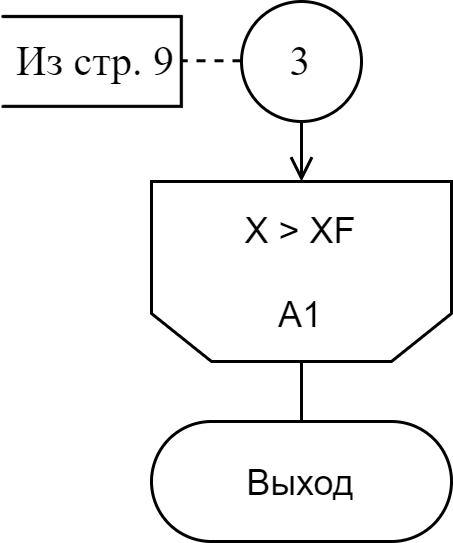


Рисунок 4 - Схема алгоритма решения задачи по ГОСТ 19.701-90 (часть 4)

# Результаты расчетов

Вследствие выполнения программы на экран выводятся следующие результаты:

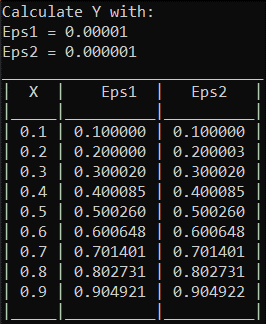


Рисунок 5 - Результаты расчётов

Приложение А

(обязательное)

program Lab5;

{

Calculate infinite function Y using stoppers

Eps1 and Eps2.

Output: X, Y with Eps1, Y with Eps2

}

{$APPTYPE CONSOLE}

{$R \*.res}

uses

System.SysUtils;

// Eps1, Eps2 - checkers to stop infinite loop

// XS - starting X

// XF - finishing X

// HX - step of X

const

Eps1 = 0.00001;

Eps2 = 0.000001;

XS = 0.1;

XF = 0.9;

HX = 0.1;

// X - argument

// Y - function

// i - count for loop

// k - count for sum

// Frac = Num / Denum

// Num = x^(4\*k+1)

// Denum = (4\*k+1)!

var

X, Y, Frac, Num: Real;

i, k, Denum: Integer;

HasReached: Boolean;

begin

writeln('Calculate Y with:');

writeln('Eps1 = ', Eps1:5:5);

writeln('Eps2 = ', Eps2:6:6);

writeln('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_');

writeln('| X | Eps1 | Eps2 |');

writeln('|\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_|');

// Set starting X

X := XS;

// Iterate through X loop

repeat

write('| ', X:1:1, ' | ');

Y := 0; // Set starting Y

Denum := 1; // Set starting denumerator

k := 0; // Set starting Sum index

Frac := X; // Calculate Fraction when k = 0

HasReached := false;

// Calculate Y

repeat

// Add fraction to sum Y

Y := Y + Frac;

// Iterate sum counter

k := k + 1;

// Calculate numerator

// Raise X in power of N using logarithmic method

// exp( N \* ln(X) )

Num := exp((4\*k+1)\*ln(X));

// Calculate factorial in denumerator

for i := 0 to 3 do

begin

Denum := Denum \* ( 4\*k + 1 - i);

end;

// Calculate fraction

Frac := Num / Denum;

// Check if fraction is smaller than Eps1

if not HasReached and (Frac <= Eps1) then

begin

HasReached := true;

// Output after Eps1 has been reached

write(Y:6:6, ' | ');

end;

until Frac <= Eps2 ;

// Output after Eps2 has been reached

writeln(Y:6:6, ' |');

// Iterate X

X := X + HX;

until X > XF;

writeln('|\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_|');

// Stop console from closing

readln;

end.

Приложение Б

(обязательное)

Тестовые наборы

Таблица - Результаты расчётов

|  |  |  |
| --- | --- | --- |
| Тест | Исходные данные и ожидаемый  результат (Mathcad) | Полученный результат |
| 1. |  |  |
| 2. |  |  |
| 3. |  |  |
| 4. |  |  |
| 5. |  |  |
| 6. |  |  |
| 7. |  |  |
| 8. |  |  |
| 9. |  |  |