**МИНОБРНАУКИ РОССИИ**

Федеральное государственное бюджетное

образовательное учреждение высшего образования

**«Ухтинский государственный технический университет»**

**(УГТУ)**

Кафедра вычислительной техники, информационных систем и технологий

**РАСЧЕТНО-ГРАФИЧЕСКАЯ РАБОТА №1**

Дисциплина: «Основы программирования»

Тема: «Простейшие конструкции языка программирования Pascal. Алгоритмы разветвленной и циклической структуры»

Шифр 191407 Группа ИСТ-2-19 Курс 1 Вариант 10

Морданов Егор Владимирович

Проверил:

старший преподаватель кафедры ВТИСиТ С. С. Сочко

Ухта

2019

содержание

[Простейшие конструкции языка программирования Pascal. Алгоритмы разветвленной и циклической структуры 3](#_Toc26635486)

[Задание 1 3](#_Toc26635487)

[Задание 2 6](#_Toc26635488)

[Задание 3 10](#_Toc26635489)

[Задание 4 14](#_Toc26635490)

[Список использованной литературы 25](#_Toc26635493)

# Простейшие конструкции языка программирования Pascal. Алгоритмы разветвленной и циклической структуры

## Задание 1

Разработать программу вычисления значения функции  
 , где , [1]

Постановка задачи

Входные данные: вещественное значение X.

Ожидаемый результат: значение функции , где , при заданном значении X с точностью до сотых.

Таблица 1.1 - Таблица тестов для задания 1

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Проверяемый случай | Входные данные | Ожидаемый результат |
| 1 | X>0 | 1 | Y = -11 |
| 2 | X<0 | -2 | Y = 37.17 |
| 3 | X=0 | 0 | Y = 0.79 |
| 4 | 0<X<1 | 0,6 | Y = -3.65 |
| 5 | 1<X<2 | 1,7 | Y = -40.71 |
| 6 | X=2 | 2 | Y не существует |



A = x2-6×x-1

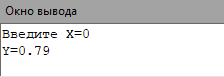
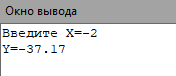
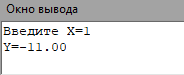
B =

Рисунок 1.1 - Блок-схема алгоритма для задания 1

Листинг программы

|  |
| --- |
| **var** a, b, x:real;  **begin**  write('Введите X=');  read(x);  a := x\*x-6\*x-1;  b := exp(ln(abs(x-2)) / 3);  **if** b = 0 **then**  writeln('Введено некорректное число')  **else**  writeln('Y=', (1+2\*a\*x)/b:0:2);  **end**. |

Протокол выполнения программы



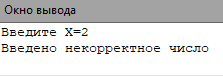
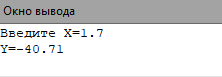
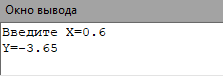


Рисунок 1.2 - Протокол выполнения программы для задания 1

## Задание 2

Дано натуральное число N. Найти максимальную цифру в записи числа и её позицию. [1]

Постановка задачи

Входные данные: натуральное число N.

Ожидаемый результат: Максимальная цифра и её позиции в записи.

Таблица 1.2 - Таблица тестов для задания 2

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Проверяемый случай | Входные данные | Ожидаемый результат |
| 1 | N>0 | 55899 | Максимальное число: 9  Позиция числа: 5,4 |
| 2 | N=5 | 5 | Максимальное число: 5  Позиция числа: 1 |
| 3 | N<0 | -55 | Повторный ввод значения |



Рисунок 1.3 - Блок-схема алгоритма для задания 2

Листинг программы

|  |
| --- |
| **var** v,p,mx,sv,i:integer;  **begin**  mx := 0;  p := 1;  i := 0;  **repeat**  write('Введите число = ');  readln(v);  **until** v > 0;  sv := v;  **while** (v > 0) **do begin**  **if** mx < v **mod** 10 **then**  mx := v **mod** 10;  v := v **div** 10;  i := i + 1;  **end**;  v := sv;  writeln('Максимальная цифра ', mx);  **while** (v > 0) **do begin**  **if** mx = v **mod** 10 **then**  writeln('Число на позиции: ',i-p+1);  v := v **div** 10;  p := p + 1;  **end**;  **end**. |

Протокол выполнения программы

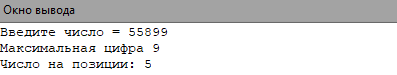




Рисунок 1.4 - Протокол выполнения программы для задания 2

## Задание 3

Разработать программу ввода данных. Количество элементов заранее неизвестно. Вводится последовательность целых чисел, в которых каждое следующее число не меньше суммы двух предыдущих. Определить количество четных элементов последовательности. [1]

Постановка задачи

Входные данные: последовательность целых чисел неизвестной длинны.

Ожидаемый результат: ввод последовательности пока выполняется условие - каждое следующее число не меньше суммы двух предыдущих; количество четных элементов последовательности.

Таблица 1.3 - Таблица тестов для задания 3

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Проверяемый случай | Входные данные | Ожидаемый результат |
| 1 | a, b, c > 0 | 1,2,3,5,8,10 | Ввод прекращается после числа 10; 2 четных элемента |
| 2 | a + b > c | 2,5,3 | Ввод прекращается после числа 3; последовательность отсутствует |
| 3 | a, b, c < 0 | -1,-2,-3,-5,-4,-1,-12 | Ввод прекращается после числа -12; 2 четных элемента |
| 4 | a, b, c – любые целые числа | 0,-6,-3,-7,9,3,7 | Ввод прекращается после числа 7; 1 четный элемент |

Рисунок 0.5 - Блок-схема алгоритма для задания 3

Начало

k=0

a

a четное

k=k+1

true

false

b

b четное

k=k+1

true

false

c

(a+b)<=c

c четное

k=k+1

true

false

a=b

b=c

c

true

false

k

(a+b)<=c

true

false

Последовательность  
отсутствует

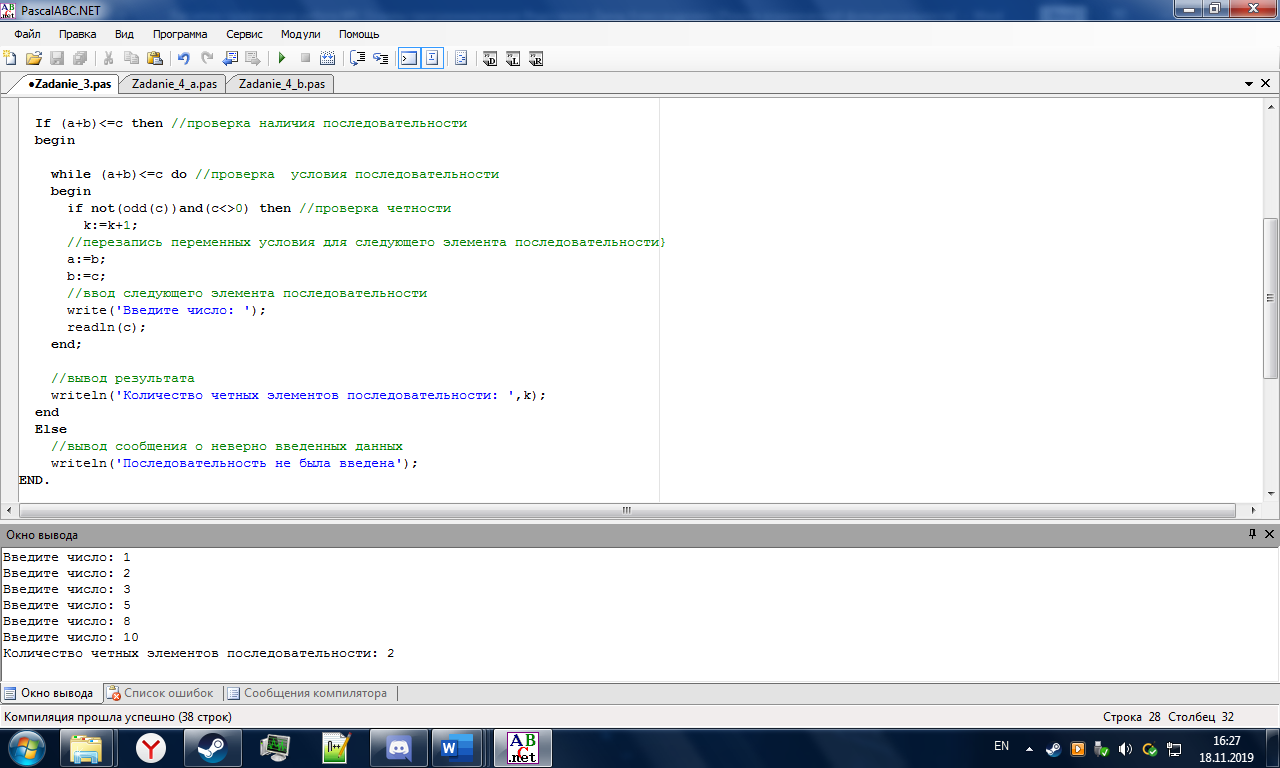
Конец

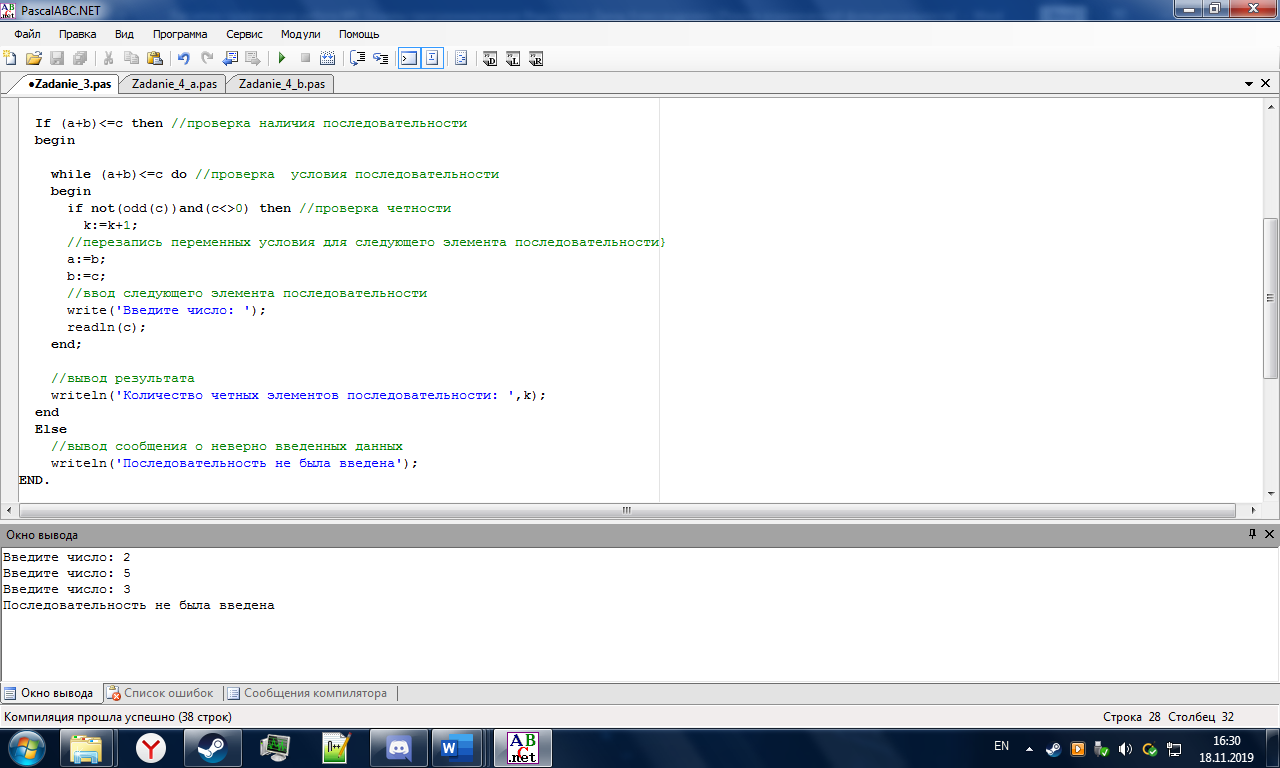
Рисунок 1.5 - Блок-схема алгоритма для задания

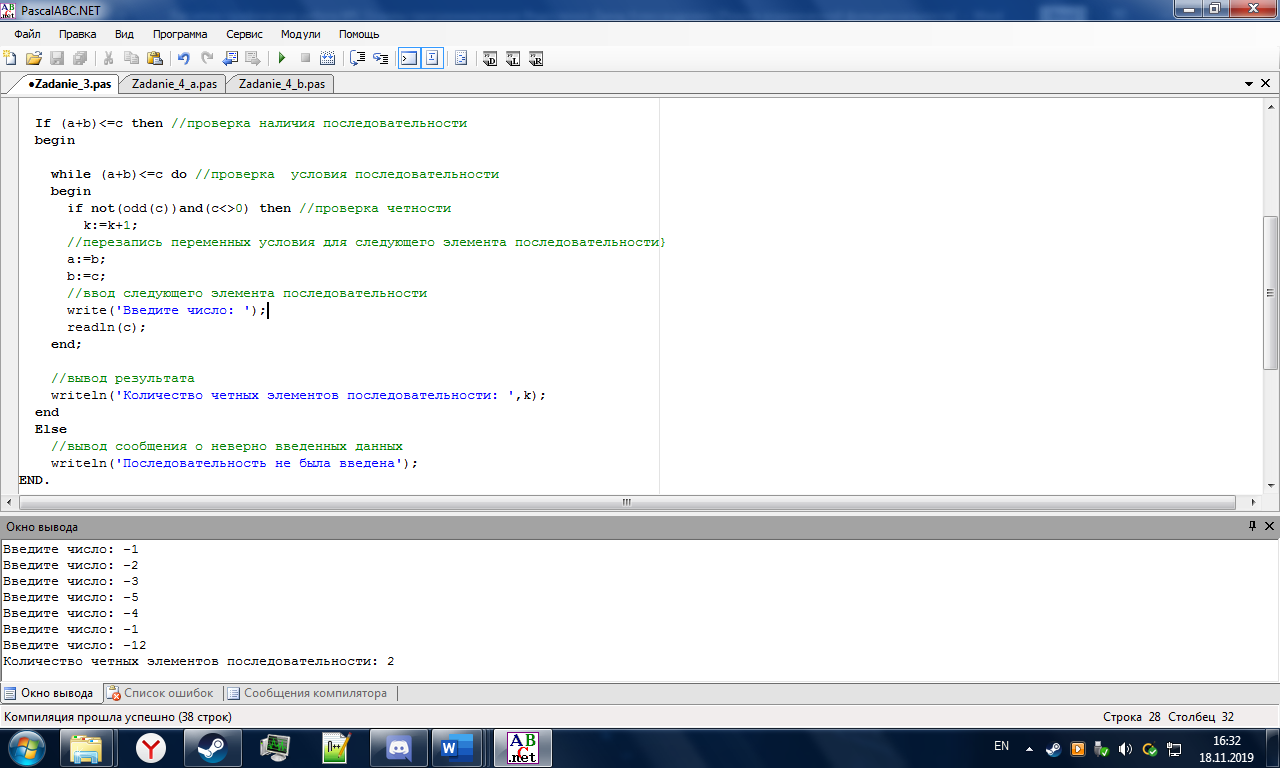
Листинг программы

|  |
| --- |
| **Var** a,b,c,k: integer; //a,b,c - числа последовательности, k - количество четных членов}  **begin**  k:=0;    //вводятся первые 3 числа и проверяется их четность  write('Введите число: ');  readln(a);  **if not**(odd(a))**and**(a<>0) **then**  k:=k+1;  write('Введите число: ');  readln(b);  **if not**(odd(b))**and**(b<>0) **then**  k:=k+1;  write('Введите число: ');  readln(c);    **If** (a+b)<=c **then begin** //проверка наличия последовательности    **while** (a+b)<=c **do** //проверка условия последовательности  **begin**  **if not**(odd(c))**and**(c<>0) **then** //проверка четности  k:=k+1;  //перезапись переменных  a:=b;  b:=c;  //ввод следующего элемента  write('Введите число: ');  readln(c);  **end**;    //вывод результата  writeln('Количество четных элементов последовательности: ',k);  **end**  **else**  //вывод сообщения об ошибке  writeln('Последовательность не была введена');  **end**. |

Протокол выполнения программы







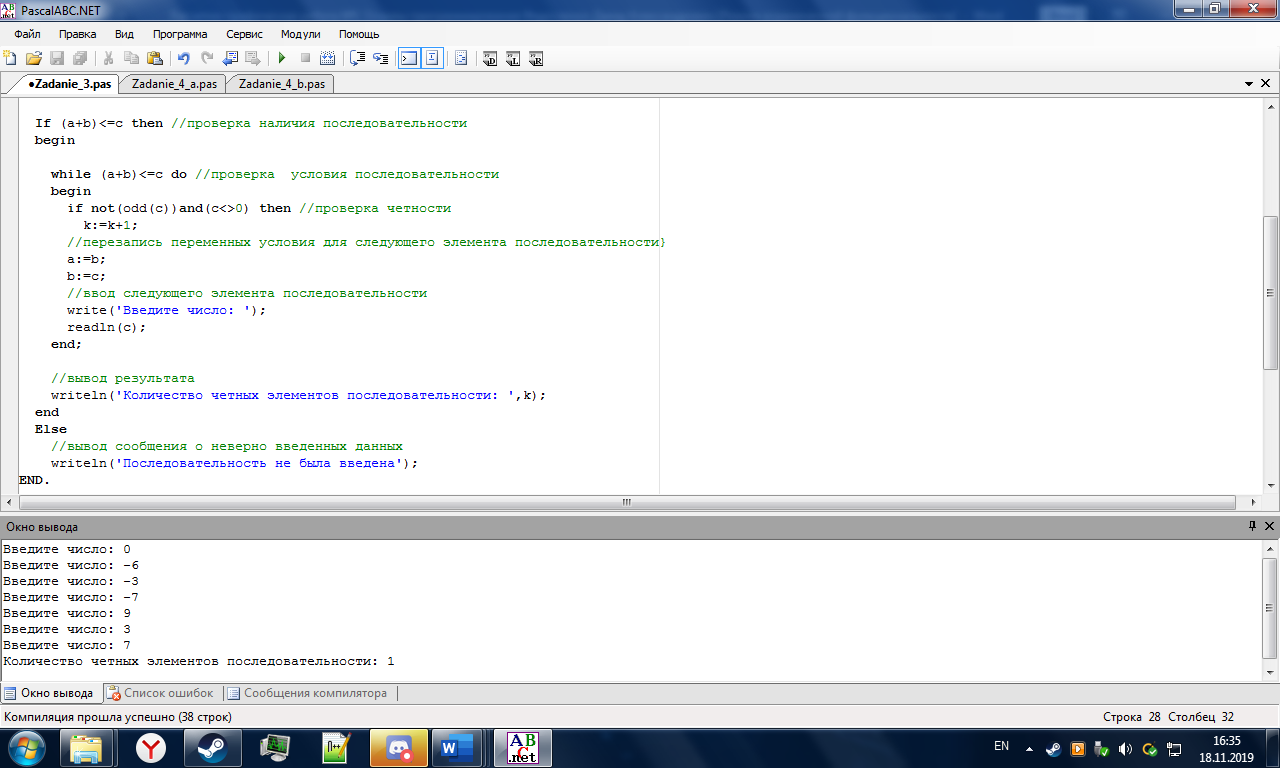


Рисунок 1.6 - Протокол выполнения программы для задания 3

## Задание 4

a) Разработать программу вычисления частичной суммы из N слагаемых ряда . [1]

Постановка задачи

Входные данные: количество слагаемых N, число X.

Ожидаемый результат: частичная сумма ряда из N слагаемых с точностью до сотых.

Таблица 1.4 - Таблица тестов для задания 4

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Проверяемый случай | Входные данные | Ожидаемый результат |
| 1 | N > 0, X > 0 | N=3, X=2 |  |
| 2 | X = 0 | N=20, X=0, |  |
| 3 | N <= 0 | N=0, X=8 | Введено неверное значение |
| 4 | -2 < X < -1 | N=5, X=-1,5 |  |
| 5 | 0 < X < 1 | N=10, X=0,5 |  |

r=0

x,n

Начало

Для i от 1 до n

r

Конец

n>0

sum = x3

Введено  
неверное  
значение

true

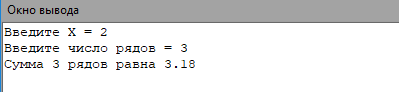
false

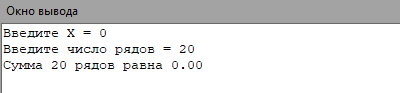
Рисунок 1.7 - Блок-схема алгоритма для задания 4 a)

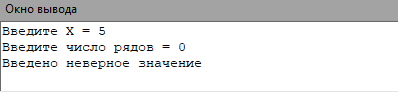
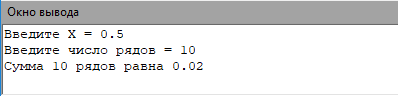
Листинг программы

|  |
| --- |
| **var** r, x, sum:real;  n,i:integer;  **begin**  r := 0;  write('Введите X = ');  read(x);  write('Введите число рядов = ');  read(n);  **if** n<=0 **then begin**  writeln('Введено неверное значение');  **exit**;  **end**;  sum:=x\*x\*x;  **for** i := 1 **to** n **do begin**  r := (r + (sum/(4\*i\*i+1)));  sum:=-1\*sum\*x\*x;  **end**;  write('Сумма ',i,' рядов равна ',r:32000:2);  **end**. |

Протокол выполнения программы







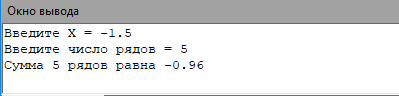


Рисунок 1.8 - Протокол выполнения программы для задания 4 a)

b) Разработать программу, вычисляющую приближенное значение суммы бесконечного ряда с точностью до ε. [1]

Постановка задачи

Входные данные: число X, точность ε.

Ожидаемый результат: приближенное значение суммы бесконечного ряда с точностью ε; количество членов ряда, вошедших в сумму.

Таблица 1.5 - Таблица тестов для задания 4

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Проверяемый случай | Входные данные | Ожидаемый результат |
| 1 | X < 1 | X = 4, E = 0,01 | Ряд стремится к бесконечности |
| 2 | X < -1 | X = -10, E = 0,01 | Ряд стремится к бесконечности |
| 3 | X = 0 | X = 0, E = 0,01 | S = 0; 1 член ряда |
| 4 | 0 < X < 1 | X = 0,5, E = 0,01 | S = 0,03; 1 члена ряда |
| 5 | -1 < X < 0 | X= -0,5, E = 0,01 | S = -0,03; 1 члена ряда |
| 6 | X = 1 | X = 1, E = 0,01 | S = 0,15; 4 член ряда |
| 7 | X = -1 | X = -1, E = 0,01 | S = -0,15; 4 член ряда |
| 8 | X = 0,25, E < 0,01 | X = 0,25, E = 0,00001 | S = 0,00307; 2 член ряда |



Рисунок 1.9 – Блок схема функция pow 4 b)



Рисунок 1.10 – Блок схема функция comma 4 b)

Начало

Конец

Рисунок 1.11 – Блок схема функция val 4 b)



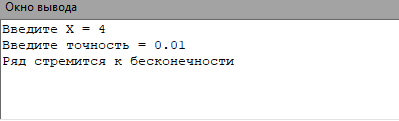
Рисунок 1.12 – Основная блок схема алгоритма 4 b)

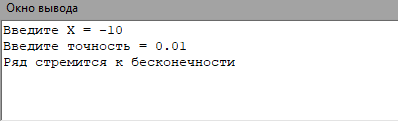
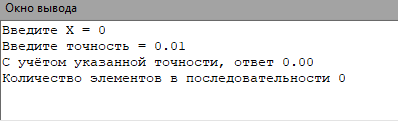
Листинг программы

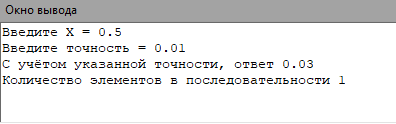
|  |
| --- |
| **var** x, cur, eps: real;  p: integer;  **function** pow(x: real; n: integer):real; **begin**  **var** res: real;  res := 1;  **while** n > 0 **do begin**  **if** n **mod** 2 = 1 **then begin**  res := res \* x;  n := n - 1;  **end**  **else begin**  x := x \* x;  n := n **div** 2;  **end**;  **end**;  pow := res;  **end**;  **function** comma(eps:real):integer; **begin**  **var** k: integer;  k:=0;  **while** eps<1 **do begin**  eps:=eps\*10;  k:=k+1;  **end**;  comma:=k;  **end**;  **function** val(i: integer): real; **begin**  val := (pow(-1, i + 1) \* (pow(x, 2\*i + 1) / (4\*i\*i + 1)));  **end**;  **begin**  write('Введите X = ');  read(x);  write('Введите точность = ');  read(eps);    **if** Abs(val(1)) < Abs(val(2)) **then begin**  writeln('Ряд стремится к бесконечности');  **exit**;  **end**;    **if** (eps <= 0) **or** (eps >= 1) **then begin**  writeln('Точность введена неправильно');  **exit**;  **end**; |

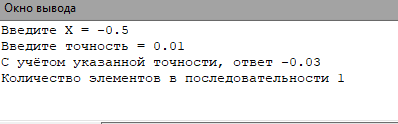
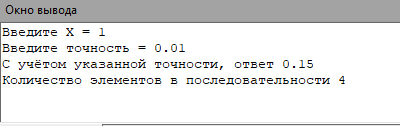
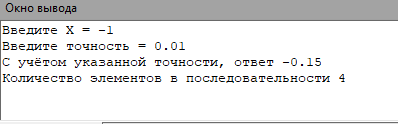
|  |
| --- |
| p := 0;  cur := 0;  **while** Abs(val(p + 1)) > eps **do begin**  p := p + 1;  cur := cur + val(p);  **end**;  **if** p = 0 **then**  p:=p + 1;  write('С учётом указанной точности, ответ ');  writeln(cur:0:comma(eps));  writeln('Количество элементов в последовательности ', p);  **end**. |

Протокол выполнения программы







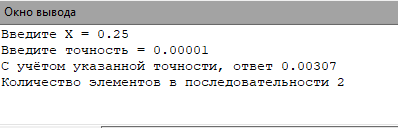


Рисунок 1.13 - Протокол выполнения программы для задания 4 b)

# Список использованной литературы

1. Центр дистанционного обучения УГТУ [Электронный ресурс]. – URL: <http://cde.ugtu.net/> (дата обращения 16.11.2019).
2. Т. Кормен, Ч. Лейзерсон, Р. Ривест К66 Алгоритмы: построение и анализ / Пер. с англ. под ред. А. Шеня – М.: МЦНМО, 2002. –960 с.: 263 ил.
3. Кудряшова, О. М. К88 Основы оформления учебных и научных работ с помощью MS Word [Текст]: методические указания к выполнению лабораторных работ по дисциплине «Программные средства общего назначения» для студентов, всех направлений квалификации Бакалавр / О. М. Кудряшова. – Ухта : УГТУ, – 28 с.