Минобрнауки РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Тульский государственный университет»

Институт прикладной математики и компьютерных наук

Кафедра информационной безопасности

**ИНФОРМАТИКА**

Отчет по выполнению лабораторной работы № 11

Вариант №15

Выполнила \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ст. гр.230711 Павлова Виктория Сергеевна

Проверила \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

доц. каф. ИБ Рудомазина Ю.Д.

Тула 2021

# Оглавление

[Оглавление 2](#_Toc88471091)

[Лабораторная работа №11 3](#_Toc88471092)

[Цель работы: 3](#_Toc88471093)

[Задания: 3](#_Toc88471094)

[Ход работы: 4](#_Toc88471095)

[Вывод: 10](#_Toc88471096)

# Лабораторная работа №11

## Цель работы:

Изучение способов организации простейших циклических алгоритмов.

## Задания:

1. Изучить теоретические положения и конспект лекций по данной теме.

2. Для любой одной программы из примера 1, приведенного в пункте «Краткие теоретические положения», составить трассировочные таблицы для значений N, равных а) 1, б) 2, в) 3. Трассировочная таблица должна иметь такое начало:

|  |  |  |  |
| --- | --- | --- | --- |
| *Трассировочная таблица для N = 1* | | | |
| ***С*** | ***n*** | ***S*** | **Пояснения** |
|  |  | 0 | *S:=0* |
|  | 1 |  | *Ввод N* |
| … | … | … | и т.д. |

Проверить работу программы для указанных значений. Сравнить результаты работы программы с рассчитанными значениями с помощью таблицы:

|  |  |  |
| --- | --- | --- |
| Значение *n* | Вычисленные значения | Результаты работы программы |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |

3. Проверить работу программы из примера 2.

4. Проверить работу программы из примера 3.

5. Для заданного варианта для каждой задачи: разработать алгоритм решения задачи. Составить программу, реализующую разработанный алгоритм. Рассчитать не менее трех контрольных значений (тестов). Проверить работу программы на рассчитанных тестах.

6. Показать результаты работы программы преподавателю.

7. Составить отчет по работе.

## Ход работы:

1. Для программы из *примера 1* *с использованием while*:

Входные данные: N, где N – целое, N>=1, а выводится сумма всех чисел от 1 до N;

**Program** Summa;

**Var** S, C, n: Integer;

**Begin**

S:=0;

ReadLn(n);

C:=1;

**While** C <= n **Do**

**Begin**

S:= S + C;

C:= C + 1;

**End**;

WriteLn('S = ', S);

**End**.

А)

|  |  |  |  |
| --- | --- | --- | --- |
| *Трассировочная таблица для N = 1* | | | |
| ***С*** | ***n*** | ***S*** | **Пояснения** |
|  |  | 0 | *S:=0* |
|  | 1 | 0 | *Ввод N* |
| 1 | 1 | 0 | *C:=1* |
| 2 | 1 | 1 | *C<=n, то*  *S:=S+C=1; С:=С+1=2;* |
|  |  | **1** | *C>n т.к 2>1, то вывод S, S=1* |

Б)

|  |  |  |  |
| --- | --- | --- | --- |
| *Трассировочная таблица для N = 2* | | | |
| ***С*** | ***n*** | ***S*** | **Пояснения** |
|  |  | 0 | *S:=0* |
|  | 2 | 0 | *Ввод N* |
| 1 | 2 | 0 | *С:=1* |
| 2 | 2 | 1 | *C<=n, верно, то*  *S:=S+C=1; С:=С+1=2;* |
| 3 | 2 | 3 | *C<=n, верно, то*  *S:=S+C=3; С:=С+1=3;* |
|  |  | **3** | *C>n т.к 3>2, то вывод S, S=3* |

В)

|  |  |  |  |
| --- | --- | --- | --- |
| *Трассировочная таблица для N = 3* | | | |
| ***С*** | ***n*** | ***S*** | **Пояснения** |
|  |  | 0 | *S:=0* |
|  | 3 | 0 | *Ввод N* |
| 1 | 3 | 0 | С:=1 |
| 2 | 3 | 1 | *C<=n,верно то*  *S:=S+C=1; С:=С+1=2;* |
| 3 | 3 | 3 | *C<=n, верно, то*  *S:=S+C=3; С:=С+1=3;* |
| 4 | 3 | 6 | *C<=n,верно, то*  *S:=S+C=6; С:=С+1=4;* |
|  |  | **6** | *C>n т.к 4>3, то вывод S, S=6* |

Расчёт значений суммы:

|  |  |  |
| --- | --- | --- |
| Значение *n* | Вычисленные значения суммы | Результаты работы программы |
| 1 | 1 | 1 |
| 2 | 1+2=3 | 3 |
| 3 | 1+2+3=6 | 6 |

|  |  |  |
| --- | --- | --- |
| Тест программы №1: N=1 | Тест программы №2: N=2 | Тест программы №3: N=3 |

Результаты работы программы сходятся с рассчитанными значениями.

1. Проверка работы программы из *примера 2.*

Входные данные: N, где N – количество чисел, целое и N>=1, X – само число (или числа), X принадлежит отрезку [-32768;32767], а выводится среднее арифметические введённых чисел.

Расчёт значений среднего арифметического:

|  |  |  |  |
| --- | --- | --- | --- |
| Кол-во чисел n | Сами числа X1, X2, …, Xn | Вычисленные значения ср. арифметического | Результаты работы программы |
| 1 | X1=1 |  | 1 |
| 2 | X1=-32768,  X2=1 |  | -16383.5 |
| 3 | X1=2000,  X2=5, X3=3567 |  | 1857.33333333333 |

|  |  |  |
| --- | --- | --- |
| Тест программы №1: N=1, X=1 | Тест программы №2: N=2,  X1=-32768, X2=1 | Тест программы №3: N=3, X1=2000, X2=5, X3=3567 |

1. Проверка работы программы из *примера 3*:

Входные данные: N, где N – количество чисел, целое и N>=0, X – само число (или числа), X принадлежит отрезку [-32768;32767], а выводится вид их последовательности (возрастающая или убывающая).

Определение вида последовательности:

|  |  |  |  |
| --- | --- | --- | --- |
| Кол-во чисел n | Сами числа X1, X2, …, Xn | Определение вида последовательности | Результаты работы программы |
| 1 | X1=1 | 1>0, последовательность возрастающая | Последовательность возрастающая |
| 2 | X1=-1000,  X2=-1002 | X2<X1, последовательность НЕ возрастающая | Последовательность не возрастающая |
| 3 | X1=2,  X2=4,  X3=6. | X3>X2>X1, последовательность возрастающая | Последовательность возрастающая |
| 4 | X1=X2=X3=X4=0 | X1=X2=X3=X4=0, последовательность НЕ возрастающая | Последовательность не возрастающая |

|  |  |  |
| --- | --- | --- |
| Тест программы №1: N=1, X=1    Тест программы №4, N=4, X1=X2=X3=X4=0 | Тест программы №2: N=2,  X1=-1000, X2=-1002 | Тест программы №3: N=3, X1=2,  X2=4, X3=6. |

1. Задачи по варианту:

Задача 1. Дано натуральное число *n*. Вычислить .   
Название файла с программой: 11lr 1zd.pas

|  |  |
| --- | --- |
| Исходные данные:  N – натуральное число, N принадлежит отрезку [1;32767]. | Выходные данные: произведение P, такое что . |

Расчёт значений произведения:

|  |  |  |
| --- | --- | --- |
| Натуральное число N: | Произведение Р: | Результаты работы программы |
| 1 | =0,(6) | 0.666666666666667 |
| 3 | \* | 0.457142857142857 |
| 5 | \* | 0.369408369408369 |

|  |  |
| --- | --- |
| Схема алгоритма:  Начало  Ввод n  P=1  Вывод p  i=1 до n  Конец | Код программы:  **begin**  write('Введите число n= ');  readln(n);  p:=1;  **for** i:=1 **to** n **do**  **begin**  p:=p\*((2\*i)/(2\*i+1));  **end**;  writeln(p);  **end**. |

Результаты работы программы:

|  |  |  |
| --- | --- | --- |
| Тест программы №1: N=1 | Тест программы №2: N=3 | Тест программы №3: N=5 |

Задача 2. С клавиатуры вводятся 15 целых чисел в диапазоне от 150 до 200 – рост студентов некоторой группы. Насколько рост самого высокого из них превышает рост самого низкого?

Название файла с программой: 11lr 2zd.pas

|  |  |
| --- | --- |
| Исходные данные:  Вводится 15 значений для X из диапазона [150;200], где X – рост студентов некоторой группы. | Выходные данные: разница в росте между самым высоким и самым низким. |

Схема алгоритма:

|  |  |
| --- | --- |
| Ввод y  y>max  max=y  да  x=1 до 15  min=200  max=150  Начало | Код программы:  **var** x,y,max,d,min: integer;  **begin**  write('Введите рост студентов не меньше 150 см и не больше 200 см =');  max:=150;  min:=200;  **for** x:=1 **to** 15 **do** //x счетчик  **begin**  readln(y); //рост людей  **if** y>max **then** max:=y;  **if** y<min **then** min:=y;  **end**;  d:=max-min;  writeln(d);  **end**. |

Конец

Вывод d

d=max-min

да

min=y

y<min

Расчёт значений произведения:

|  |  |  |
| --- | --- | --- |
| Значения роста (см) | Вычисление разницы между самым высоким и низким (см) | Результаты работы программы (см) |
| Все одного роста: Y1=...=Y15=150 | *Разница в росте отсутствует (0)* | 0 |
| Y1=167, Y2=200,  Y3=…=Y15=160 | *Минимум=160, максимум=200,*  *разница 200-160=40.* | 40 |
| Y1=161,Y2=162, …, Y15=175 | *Минимум=161, максимум=175,*  *разница 175-161=14.* | 14 |

Результаты работы программы:

|  |  |  |
| --- | --- | --- |
| Тест программы №1: N=1 | Тест программы №2: N=3 | Тест программы №3: N=5 |

## Вывод:

Я познакомилась со способом организации простейших циклических алгоритмов.