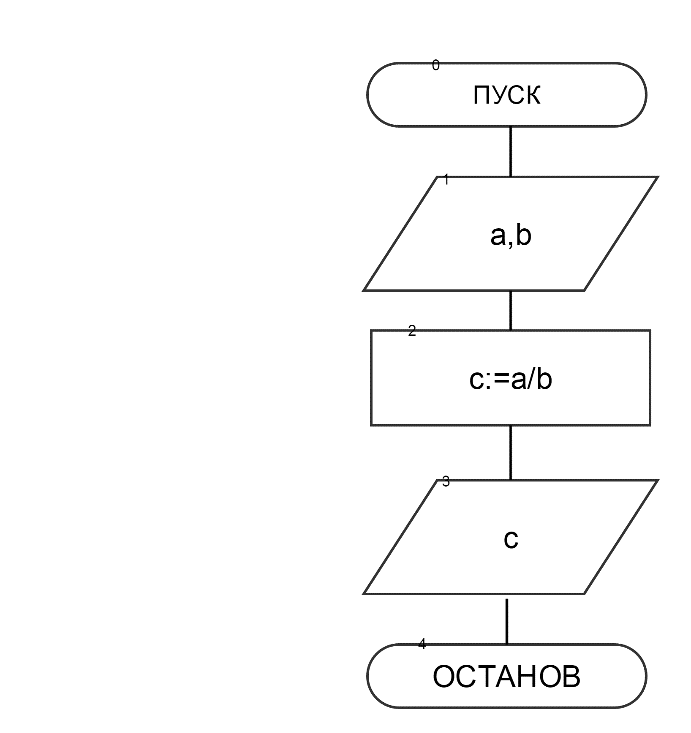
Лабораторная работа №1.

1. Тема: Линейно-вычислительный процесс.
2. Цель: научиться реализовывать алгоритмы линейно-вычислительных процессов средствами Free Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: с клавиатуры вводятся 2 числа, найти их частное.
5. Математическая модель: c=a:b
6. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| a | real | делимое |
| b | real | делитель |
| c | real | частное |

1. Код программы: program lrchastnoe; var a,b,c: real; begin Writeln(‘Vvedite Delimoe: ’); readln(a); Writeln(‘Vvedite Delitel ne ravniy 0: ’);

Readln(b);

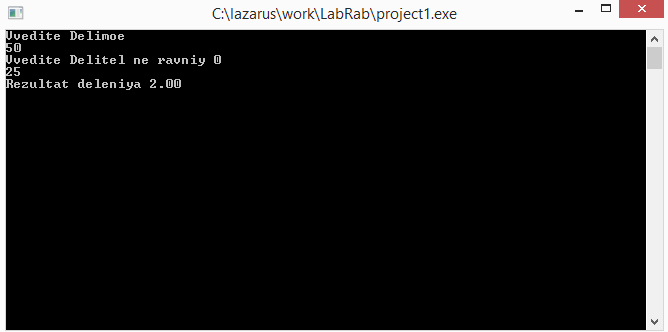
c:=a/b;

Writeln(‘Rezultat deleniya: ’,c:2:3);

Readln;

End.

1. Результат работы программы:



1. Анализ результатов вычисления:

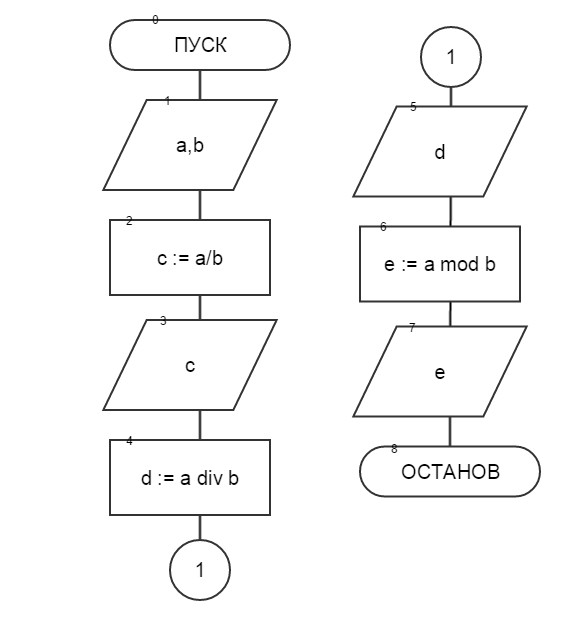
Наша программа выводит результат деления числа А на число В, но она не предусматривает, что делить на ноль нельзя и если пользователь введёт вместо числа В ноль, то программа выдаст ошибку. (Кроме того, мы вводим русский текст латинскими буквами – а ведь это довольно просто исправить.)

1. Вывод:

Мы написали простейшую программу для нахождения частного, входными данными для которой являются два числа, вводимые с клавиатуры. Однако наша программа нуждается в доработке.

**Задание 1.**

1. Тема: Линейные вычислительные процессы.
2. Цель: научиться реализовывать алгоритмы линейно-вычислительных процессов средствами Free Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: даны числа 7 и 5, определить результат вещественного деления, целочисленного деления и найти остаток от целочисленного деления.
5. Математическая модель: c = 7 : 5; d = 7 div 5; e = 7 mod 5.
6. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| a | const | делимое |
| b | const | делитель |
| c | real | частное |
| d | integer | частное |
| e | integer | частное |

1. Код программы:

program bahurev\_nikolay\_lr1;

const

a = 7;

b = 5;

var

c : real;

d, e : integer;

begin

c := a/b;

writeln('Rezultat Veshestvennogo Deleniya: ',c:2:2);

d := a div b;

writeln('Rezultat Celochislennogo Deleniya: ',d);

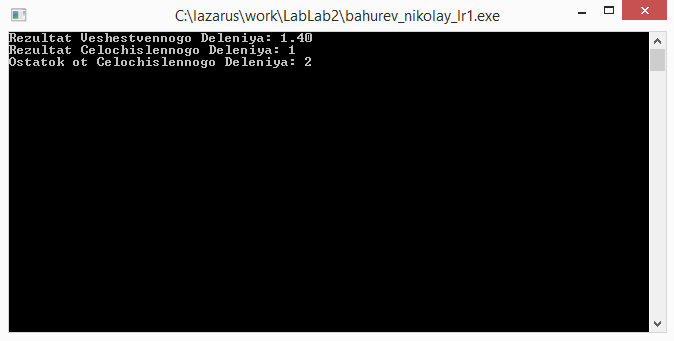
e := a mod b;

writeln('Ostatok ot Celochislennogo Deleniya: ',e);

readln();

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

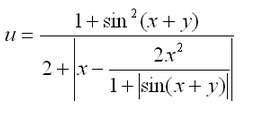
Наша программа выводит три числа: результат вещественного деления 7 на 5, результат целочисленного деления и остаток от целочисленного деления. Программа призвана выполнить только эти вычисления, поэтому числа 7 и 5 не вводятся пользователем, а прописаны в программе в качестве констант.

1. Вывод:

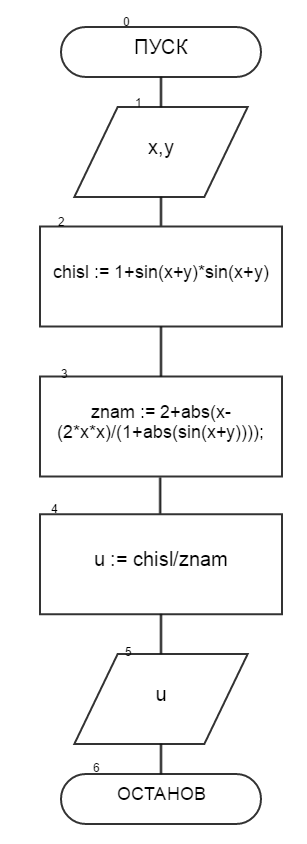
Нами написана программа, которая выводит результат вещественного деления, целочисленного деления и остаток от целочисленного деления 7 на 5.

**Задание 2.**

1. Тема: Линейные вычислительные процессы.
2. Цель: научиться реализовывать алгоритмы линейно-вычислительных процессов средствами Free Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: написать программу, в качестве вводных данных для которой будут числа X и Y. Найти значение переменной U (найти значение данного выражения).
5. Математическая модель:



1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| u | real | Значение выражения |
| x | real | Вводные данные |
| y | real | Вводные данные |
| chisl | real | Числитель дроби |
| znam | real | Знаменатель дроби |

1. Код программы:

program bahurev\_nikolay\_lr2\_pr;

var

u,x,y,chisl,znam : real;

begin

writeln('Vvedite X: ');

readln(x);

writeln('Vvedite Y: ');

readln(y);

begin

chisl := 1+sin(x+y)\*sin(x+y);

znam := 2+abs(x-(2\*x\*x)/(1+abs(sin(x+y))));

u := chisl/znam;

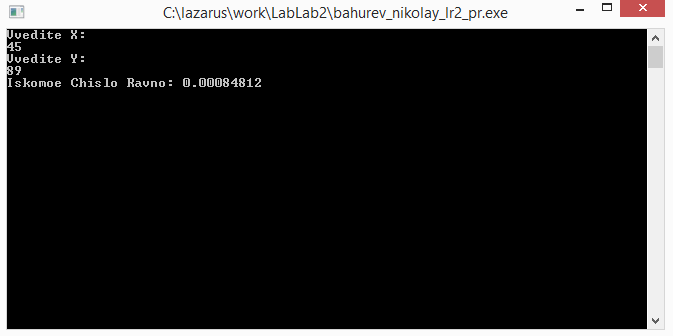
writeln('Iskomoe Chislo Ravno: ',u:3:8);

end;

readln();

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

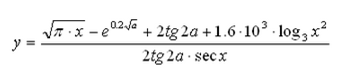
Пользователь ввёл два числа: X и Y. Для удобства вычислительный процесс разбит на несколько шагов. Сначала программа находит значение выражения в числителе дроби, потом в знаменателе и только потом, непосредственно, считает саму дробь.

1. Вывод:

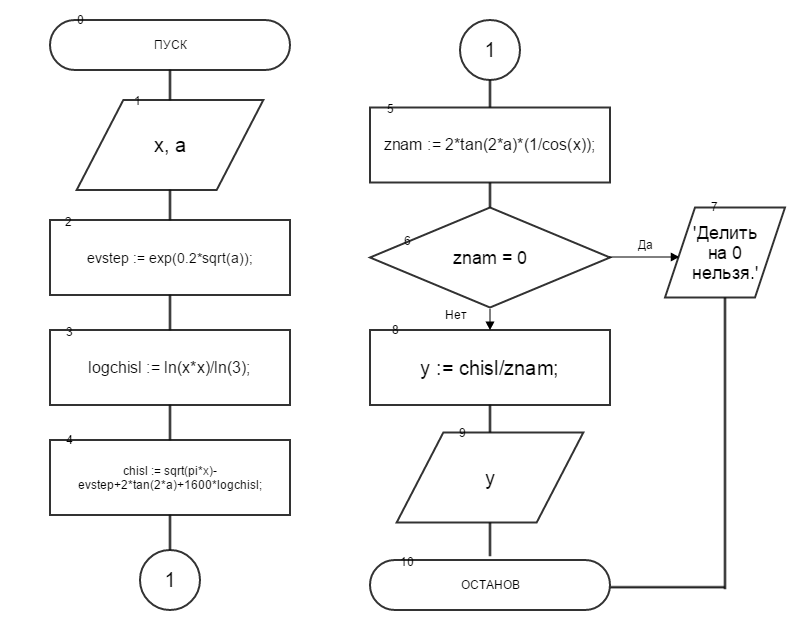
Нами написана программа, которая, получив на ввод два числа X и Y высчитывает по формуле значение U.

**Задание 3.**

1. Тема: Линейные вычислительные процессы.
2. Цель: научиться реализовывать алгоритмы линейно-вычислительных процессов средствами Free Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: написать программу, в качестве вводных данных для которой будут числа X и A. Найти значение переменной Y.
5. Математическая модель:



1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| a | real | Вводные данные |
| x | real | Вводные данные |
| y | real | Искомое число |
| chisl | real | Числитель дроби |
| znam | real | Знаменатель дроби |
| evstep | real | Возведение в степень |
| logchisl | real | Вычисление логарифма |

1. Код программы:

program bahurev\_nikolay\_lr3;

var

a,x,y,

chisl,znam,

evstep,

logchisl : real;

begin

writeln('Vvedite X: ');

readln(x);

writeln('Vvedite A: ');

readln(a);

evstep := exp(0.2\*sqrt(a));

logchisl := ln(x\*x)/ln(3);

chisl := sqrt(pi\*x)-evstep+2\*(sin(2\*a)/cos(2\*a))+1600\*logchisl;

znam := 2\*(sin(2\*a)/cos(2\*a))\*(1/cos(x));

if znam = 0 then

writeln('Delit na 0 nelzy.') else

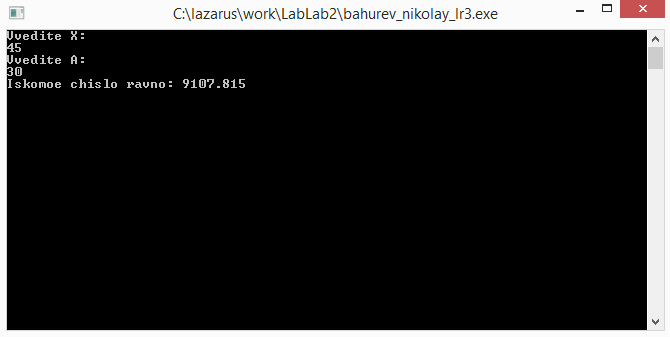
y := chisl/znam;

writeln('Iskomoe chislo ravno: ',y:3:3);

readln();

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

Результатом работы программы является вещественное число, получаемое путем обработки входных данных по заданной формуле.

1. Вывод:

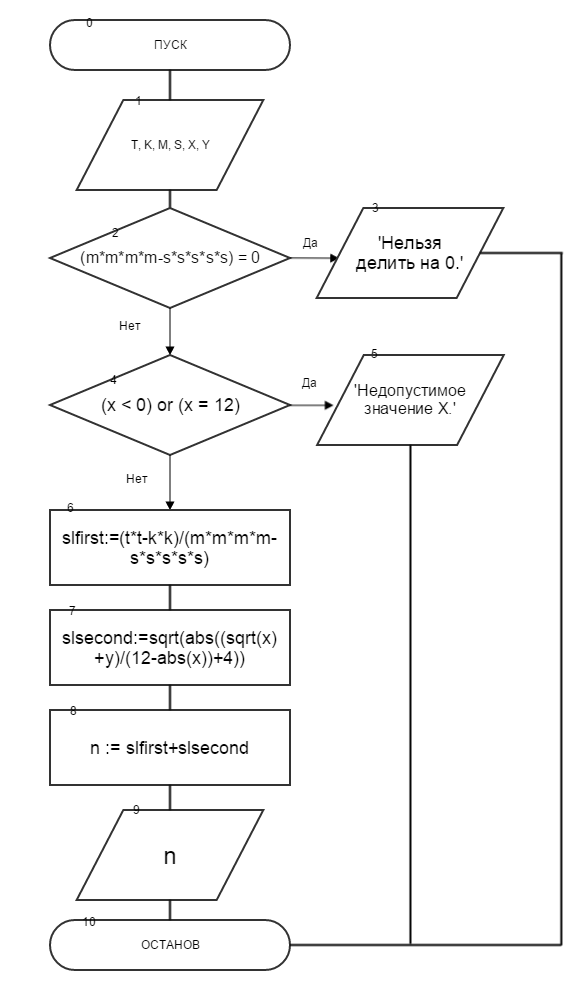
Мы написали программу, в которую пользователь вводит числа X и A, после чего программа вычисляет значение переменной Y. Как и в предыдущей программе процесс вычисления ради удобства разбит на несколько действий.

**Задание 4.**

1. Тема: Линейные вычислительные процессы.
2. Цель: научиться реализовывать алгоритмы линейно-вычислительных процессов средствами Free Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: написать программу, которая вычисляет значение выражения по формуле. В качестве входных данных выступают числа T, K, M, S, X, Y.
5. Математическая модель:



1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| y | real | Вводные данные |
| x | real | Вводные данные |
| n | real | Искомое число |
| slfirst | real | Первое слагаемое |
| slsecond | real | Второе слагаемое |
| t | real | Вводные данные |
| k | real | Вводные данные |
| m | real | Вводные данные |
| s | real | Вводные данные |

1. Код программы:

program bahurev\_nikolay\_lr4;

var

n,x,y,

slfirst,slsecond,

t,k,m,s : real;

begin

writeln('Vvedite T: ');

readln(t);

writeln('Vvedite K: ');

readln(k);

writeln('Vvedite M: ');

readln(m);

writeln('Vvedite S: ');

readln(s);

writeln('Vvedite X: ');

readln(x);

writeln('Vvedite Y: ');

readln(y);

if (m\*m\*m\*m-s\*s\*s\*s\*s) = 0 then

writeln('Nelzya delit na 0.') else

if (x < 0) or (x = 12) then

writeln('Nedopustimoe znachenie X.') else begin

slfirst := (t\*t-k\*k)/(m\*m\*m\*m-s\*s\*s\*s\*s);

slsecond := sqrt(abs((sqrt(x)+y)/(12-abs(x))+4));

n := slfirst+slsecond;

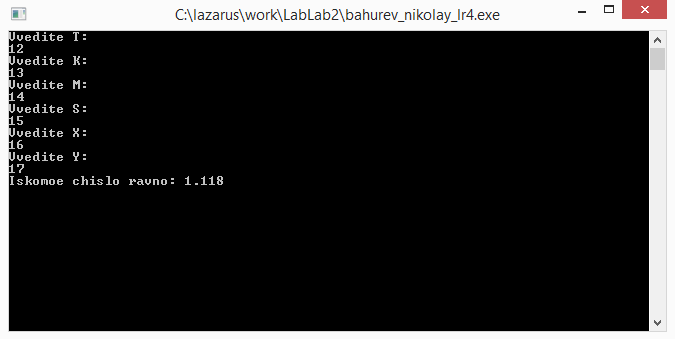
writeln('Iskomoe chislo ravno: ',n:3:3);

end;

readln();

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

Программа выдаёт в качестве решения вещественное число, предварительно проводя проверку входных данных. Она проверяет, не равняется ли знаменатель первой дроби 0, не отрицательно ли подкоренное выражение и не равняется ли 0 знаменатель второй дроби.

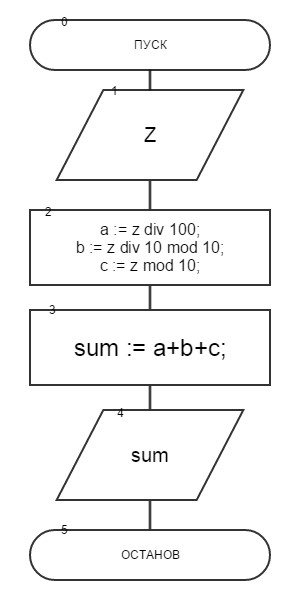
1. Вывод: мы написали программу, в которую пользователь вводит числа T, K, M, S, X, Y. Затем программа по заданной формуле находит значение выражения (переменная n).

**Задание 5.**

1. Тема: Линейные вычислительные процессы.
2. Цель: научиться реализовывать алгоритмы линейно-вычислительных процессов средствами Free Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: написать программу, в которую с клавиатуры вводится трехзначное число и она вычисляет сумму цифр этого числа.
5. Математическая модель: a = z **div** 100; b = z **div** 10 **mod;** c = z **mod** 10;

sum = a+b+c;

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| a | integer | Первая цифра числа |
| b | integer | Вторая цифра числа |
| c | integer | Третья цифра числа |
| z | integer | Вводимое число |
| sum | integer | Сумма цифр числа |

1. Код программы:

program bahurev\_nikolay\_lr5;

var

z,a,b,c,sum : integer;

begin

writeln('Vvedite trehznachnoe chislo: ');

readln(z);

a := z div 100;

b := z div 10 mod 10;

c := z mod 10;

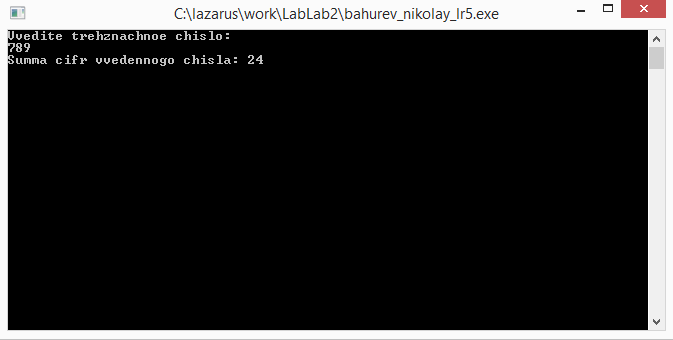
sum := a+b+c;

writeln('Summa cifr vvedennogo chisla: ',sum);

readln();

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

Программа разбивает число на разряды: сотни, десятки, единицы. После чего складывает значения полученных «цифр».

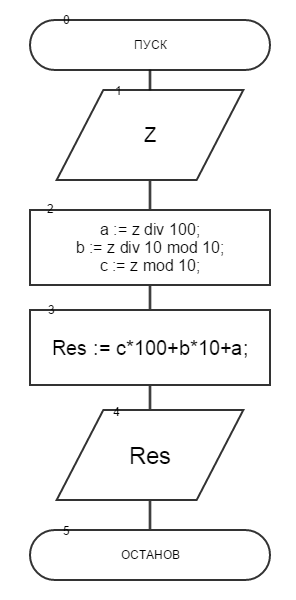
1. Вывод: мы написали программу, в которую пользователь вводит трёхзначное число и получает на выходе сумму цифр этого числа.

**Задание 6.**

1. Тема: Линейные вычислительные процессы.
2. Цель: научиться реализовывать алгоритмы линейно-вычислительных процессов средствами Free Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: написать программу, в которую с клавиатуры вводится трехзначное число и она меняет крайние цифры этого числа местами.
5. Математическая модель: a = z **div** 100; b = z **div** 10 **mod** 10; c = z **mod** 10;

Res = c\*100+b\*10+a;

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| a | integer | Первая цифра числа |
| b | integer | Вторая цифра числа |
| c | integer | Третья цифра числа |
| z | integer | Вводимое число |
| Res | integer | Полученное число |

1. Код программы:

program bahurev\_nikolay\_lr6;

var

z,a,b,c,Res : integer;

begin

writeln('Vvedite trehznachnoe chislo: ');

readln(z);

a := z div 100;

b := z div 10 mod 10;

c := z mod 10;

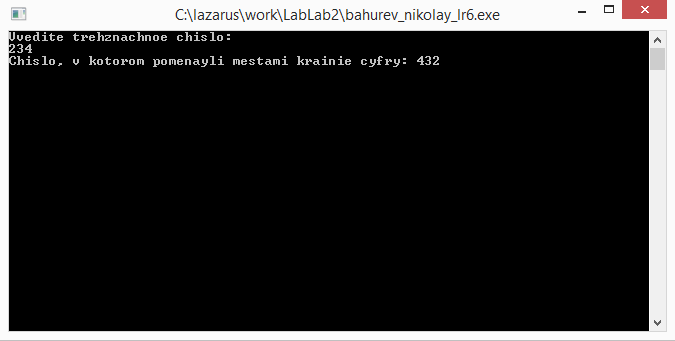
Res := c\*100+b\*10+a;

writeln('Chislo, v kotorom pomenayli mestami krainie cyfry: ',Res);

readln();

end.

1. Результат работы программы:



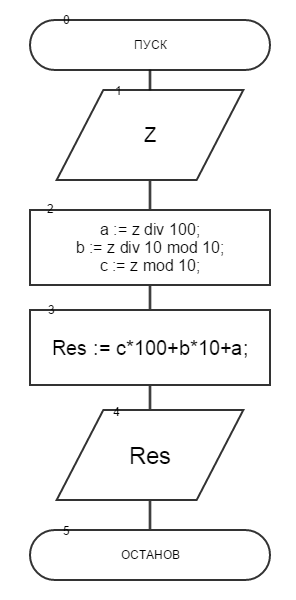
1. Анализ результатов вычисления:

Программа разбивает число на разряды: сотни, десятки, единицы. После чего вновь «собирает» число, но уже поменяв местами разряды сотен и единиц.

1. Вывод: мы написали программу, в которую пользователь вводит трёхзначное число и получает на выходе число в котором поменяли местами крайние цифры.

**Задание 7.**

1. Тема: Линейные вычислительные процессы.
2. Цель: научиться реализовывать алгоритмы линейно-вычислительных процессов средствами Free Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: написать программу, которая ищет количество тысяч, десятков и единиц в заданном числе.
5. Математическая модель: a = z **div** 1000; c = z **mod** 100 **div** 10; z = z **mod** 10;
6. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| a | integer | Кол-во тысяч |
| z | integer | Входные данные, кол-во единиц |
| c | integer | Кол-во десятков |

1. Код программы:

program bahurev\_nikolay\_lr7;

var

z,a,c : integer;

begin

writeln('Vvedite chetyrehznachnoe chislo: ');

readln(z);

a := z div 1000;

c := z mod 100 div 10;

z := z mod 10;

writeln('Kolichestvo tysych: ',a);

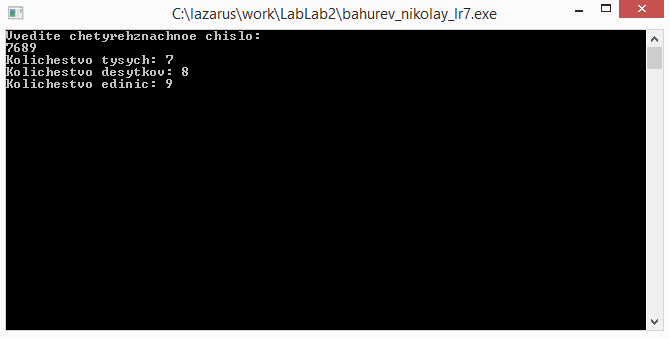
writeln('Kolichestvo desytkov: ',c);

writeln('Kolichestvo edinic: ',z);

readln();

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

Программа разбивает число на разряды: тысячи, десятки, единицы. После чего выводит на экран значение каждого разряда.

1. Вывод: мы написали программу, в которую пользователь вводит четырёхзначное число и получает на выходе количество тысяч, десятков и единиц.

**Задание 8.**

1. Тема: Линейные вычислительные процессы.
2. Цель: научиться реализовывать алгоритмы линейно-вычислительных процессов средствами Free Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: написать программу, которая определяет сколько полных часов и полных минут прошло от начала суток к заданному моменту, который определяется количеством секунд, что ввёл пользователь.
5. Математическая модель:

Пока секунды >= 60

Секунда = секунда - 60

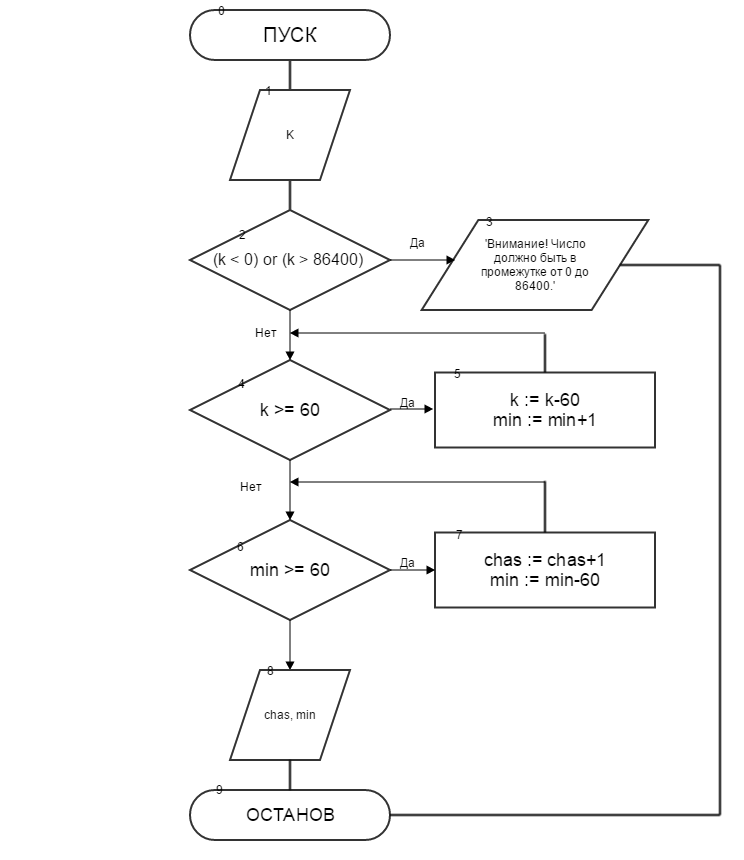
Минуты = минуты + 1

Пока минуты >= 60

Час = час +1

Минуты = 0

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| chas | longint | Кол-во часов |
| min | longint | Кол-во минут |
| k | longint | Входные данные (кол-во секунд) |

1. Код программы:

program bahurev\_nikolay\_lr8;

var

min,chas,k : longint;

begin

chas:= 0;

min:= 0;

writeln('Vvedite chislo ot 0 do 86400: ');

readln(k);

if (k < 0) or (k > 86400) then

writeln('Vnimanie! Chislo doljno byt v promejutke 0 do 86400.') else

while k >= 60 do begin

k := k-60;

min := min+1;

while min >= 60 do begin

chas := chas+1;

min := min-60;

end;

end;

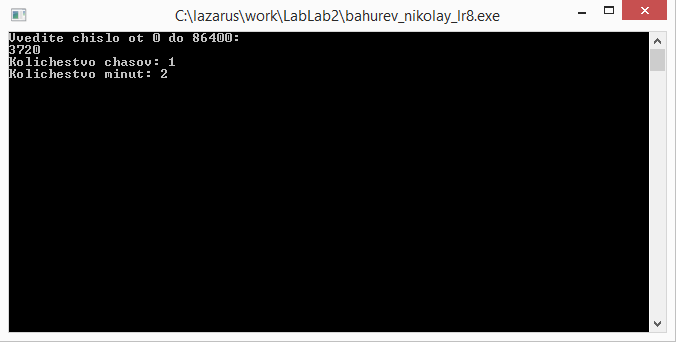
writeln('Kolichestvo chasov: ',chas);

writeln('Kolichestvo minut: ',min);

readln();

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

Наша программа проверяет больше ли кол-во секунд 60 и если больше, то прибавляет к кол-ву минут единицу и отнимает 60 от кол-ва секунд. Аналогично для кол-ва часов. В итоге программа выводит кол-во часов и минут, прошедших с начала суток до момента, введённого пользователем.

1. Вывод: нами написана программа, которая определяет сколько полных часов и полных минут прошло от начала суток к заданному моменту, который определяется количеством секунд, что ввёл пользователь.

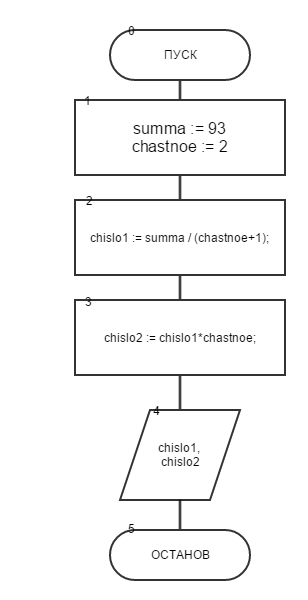
**Задание 9.**

1. Тема: Линейные вычислительные процессы.
2. Цель: научиться реализовывать алгоритмы линейно-вычислительных процессов средствами Free Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: написать программу, которая определит пару чисел, если их сумма равна 93 и одно из чисел больше второго в два раза.
5. Математическая модель:

Первое число = sum/(chastnoe+1)

Второе число = Первое число \* 2

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| summa | real | Сумма чисел |
| chastnoe | real | Частность чисел |
| chislo1 | real | Первое число |
| chislo2 | real | Второе число |

1. Код программы:

program bahurev\_nikolay\_lr9;

var

summa,

chastnoe,

chislo1,

chislo2:real;

begin

summa := 93;

chastnoe := 2;

chislo1 := summa/(chastnoe+1);

chislo2 := chislo1\*chastnoe;

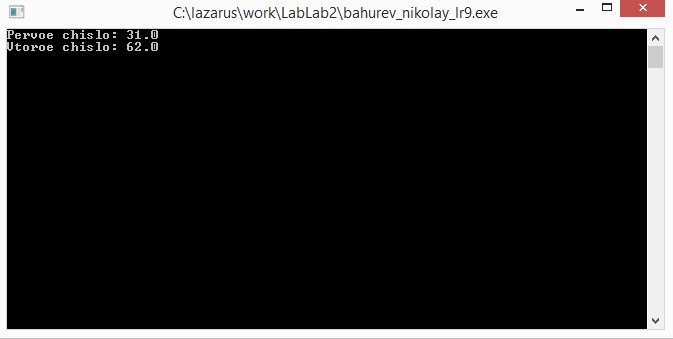
writeln('Pervoe chislo: ', chislo1:3:1);

writeln('Vtoroe chislo: ', chislo2:3:1);

readln();

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

Наша программа выполняет вычисления конкретно для заданных параметров, поэтому им автоматически присваиваются нужные значения. В результате решения линейного уравнения мы находим первое число, затем, зная, что оно меньше второго числа, находим второе умножением на 2.

1. Вывод: нами написана программа, которая определит пару чисел, если их сумма равна 93 и одно из чисел больше второго в два раза.