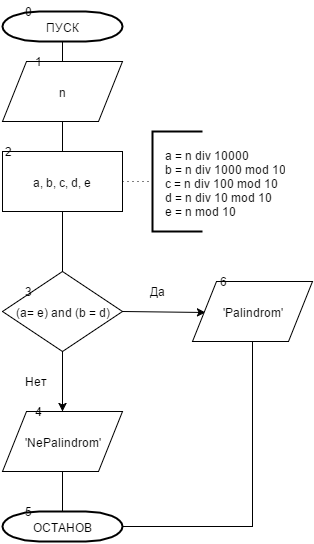
Лабораторная работа №9.

Задание 1.

1. Тема: разветвляющиеся вычислительные процессы. Оператор выбора
2. Цель: научиться реализовывать разветвляющиеся вычислительные процессы средствами FreePascal
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: gрограмма считывает с клавиатуры пятизначное число и определяет, является ли оно палиндромом.
5. Математическая модель: разбиваем число на разряды и проверяем равно ли значение первого разряда последнему, а второе – четвертому.
6. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| n | integer | Вводимое число |
| a | integer | Разряд десятков тысяч |
| b | integer | Разряд тысяч |
| c | integer | Разряд сотен |
| d | integer | Разряд десятков |
| e | integer | Разряд единиц |

1. Код программы:

program lr9task1;

var n, a,b,c,d,e : integer;

begin

writeln('Vvedite 5-ti znachnoe chislo: '); readln(n);

a := n div 10000;

b := n div 1000 mod 10;

c := n div 100 mod 10;

d := n div 10 mod 10;

e := n mod 10;

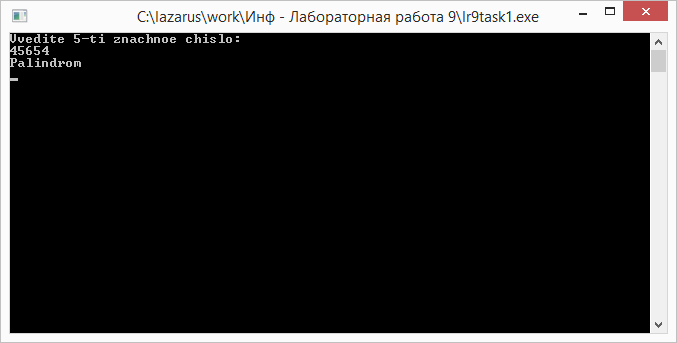
if (a = e) and (b = d) then writeln('Palindrom')

else writeln('Ne Palindrom');

readln;

end.

1. Результат работы программы:



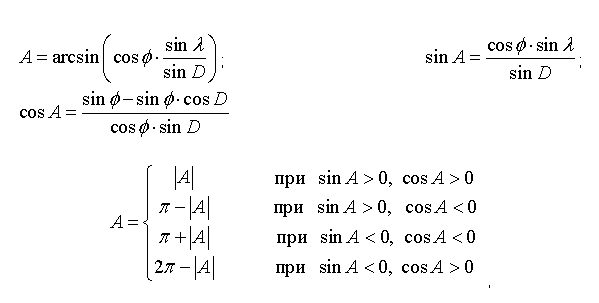
1. Анализ результатов вычисления:

Поочередно разбиваем число по разрядам. Потом с помощью проверки условия: первый разряд равен последнему и второй предпоследнему определяем является ли число палиндромом.

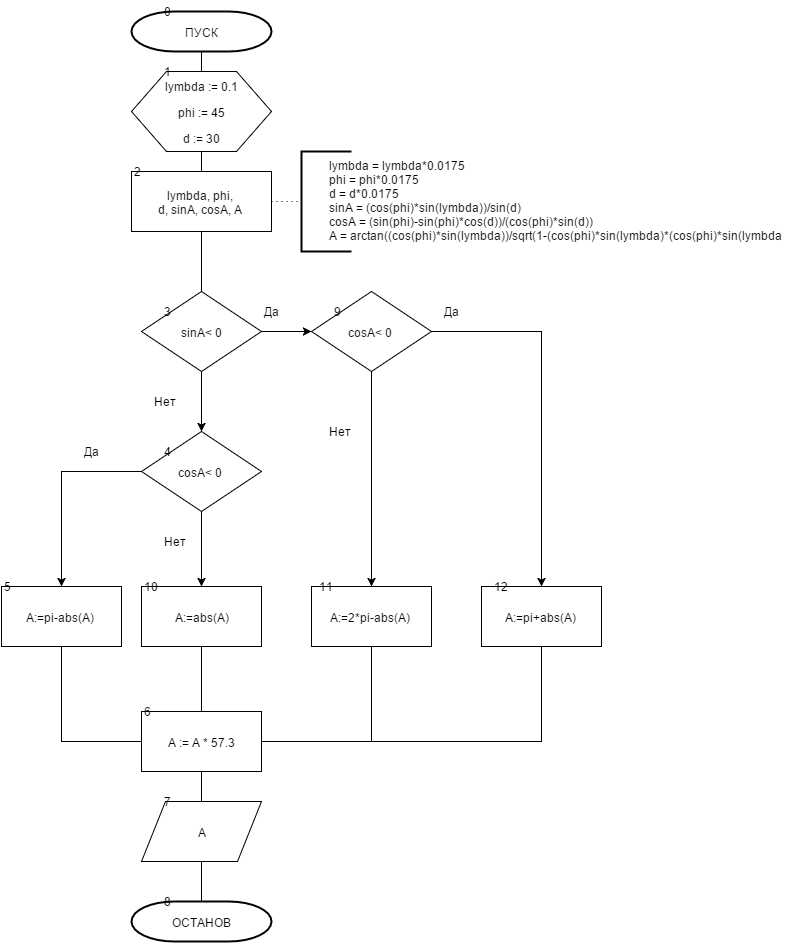
1. Вывод: программа работает корректно.

Задание 2.

1. Тема: разветвляющиеся вычислительные процессы. Оператор выбора
2. Цель: научиться реализовывать разветвляющиеся вычислительные процессы средствами FreePascal
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: определить четверть угла азимута А судна. Где λ = 0.1, D = 30, φ = 45
5. Математическая модель:



1. Блок-схема: см. на следующей странице



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| lymbda | real |  |
| phi | real |  |
| d | real |  |
| sinA | real |  |
| cosA | real |  |
| A | real |  |

1. Код программы:

program task1;

var

lymbda, phi, d, sinA, cosA, A: real;

begin

lymbda := 0.1;

phi := 45;

d := 30;

lymbda := lymbda\*0.0175;

phi := phi\*0.0175;

d := d\*0.0175;

sinA := (cos(phi)\*sin(lymbda))/sin(d);

cosA := (sin(phi)-sin(phi)\*cos(d))/(cos(phi)\*sin(d));

A := arctan((cos(phi)\*sin(lymbda))/sqrt(1-(cos(phi)\*sin(lymbda)\*(cos(phi)\*sin(lymbda)))));

if sinA < 0 then

if cosA < 0 then A:=pi+abs(A) else A:=2\*pi-abs(A) else

if cosA < 0 then A:=pi-abs(A) else A:=abs(A);

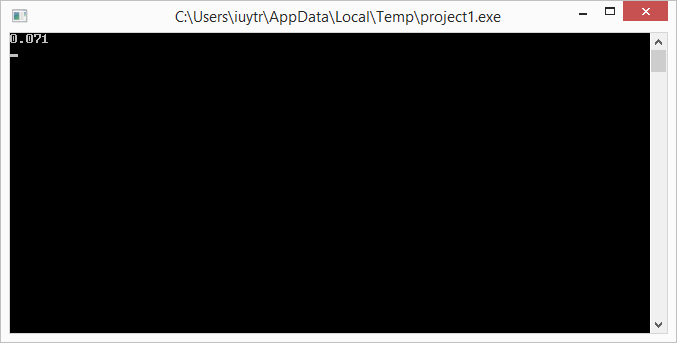
A := A \* 57.3;

writeln(A:2:3);

readln();

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

Мы проводим вычисления по формулам, предварительно переведя градусы в радианы. Затем начинаем проверку всех условий и выбор соответствующего оператора. После чего переводим значение искомой переменной из радианов в градусы.

1. Вывод: программа работает корректно.

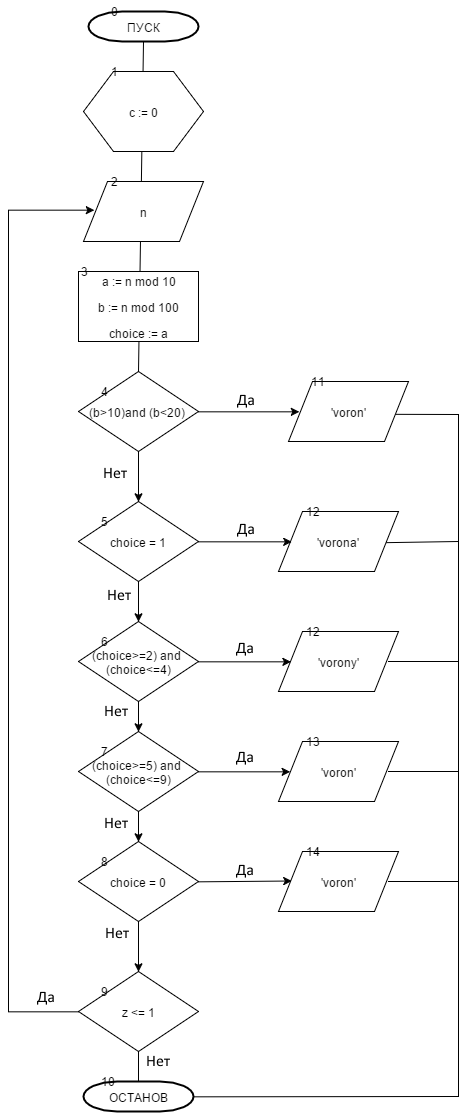
Задание 3.

1. Тема: разветвляющиеся вычислительные процессы. Оператор выбора
2. Цель: научиться реализовывать разветвляющиеся вычислительные процессы средствами FreePascal
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: сформировать вывод слова «ворона» в зависимости от любого числительного, которое вводится с клавиатуры.
5. Математическая модель:

мы проверяем последние две цифры. Если они образуют число больше 10 и меньше 20, то пишем «ворон». Потом проверяем последнюю цифру. Если 1, то «ворон»,

если от 2 до 4, то «вороны», если от 5 до 9 или 0, то «ворон».

1. Блок-схема: см. на следующей странице



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| n | integer | Кол-во ворон |
| z | integer | Организация ввода |
| a | integer | Проверка условия |
| b | integer | Проверка условия |
| choice | integer | Оператор выбора |

1. Код программы:

program lr9task3;

var

n,z,a,b,choice : integer;

begin

z := 0;

repeat

readln(n);

a := n mod 10;

b := n mod 100;

If (b>10) and (b<20) then write('voron')

else begin

choice := a;

case choice of

1 : writeln('vorona');

2..4: writeln('vorony');

5..9,0: writeln('voron');

end;

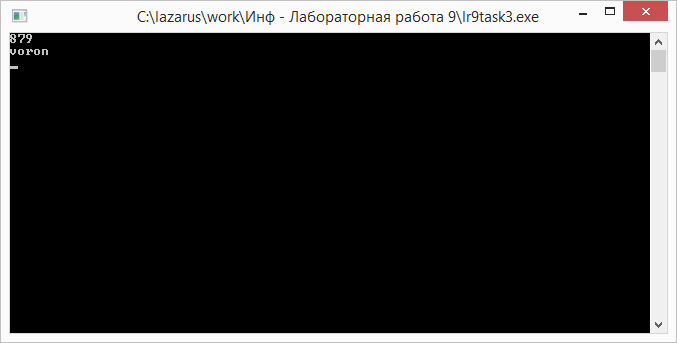
end;

readln();

until z > 1;

end.

1. Результат работы программы:



1. Анализ результатов вычисления: мы проверяем последние две цифры. Если они образуют число больше 10 и меньше 20, то пишем «ворон». Потом проверяем последнюю цифру. Если 1, то «ворон»,

если от 2 до 4, то «вороны», если от 5 до 9 или 0, то «ворон». Продолжаем программу для удобства проверки.

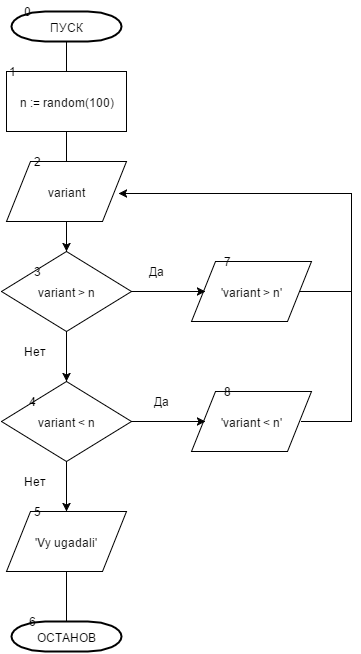
1. Вывод: программа работает корректно.

Задание 4.

1. Тема: разветвляющиеся вычислительные процессы. Оператор выбора
2. Цель: научиться реализовывать разветвляющиеся вычислительные процессы средствами FreePascal
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: составить программу, которая бы случайным образом загадывала число от 1 до 100, и предлагала Вам его угадать. При неправильном ответе, программа должна выводить сообщение о том, больше загаданное число или меньше. В случае победы выводится поздравление. Программа дает возможность вводить число до тех пор, пока пользователь не угадает.
5. Математическая модель:

Если наш вариант больше или меньше заданного числа, программа должна вывести оповещение. Когда мы угадаем, она должна сказать нам об этом.

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| n | integer | Случайное число |
| variant | integer | Вариант пользователя |

1. Код программы:

program lr9task4;

var

n, variant : integer;

begin

randomize;

n := random(100);

writeln('Ugadaite chislo ot 0 do 100');

repeat

writeln('Vvedite variant: '); readln(variant);

if (variant > n) then writeln('variant > n');

if (variant < n) then writeln('variant < n');

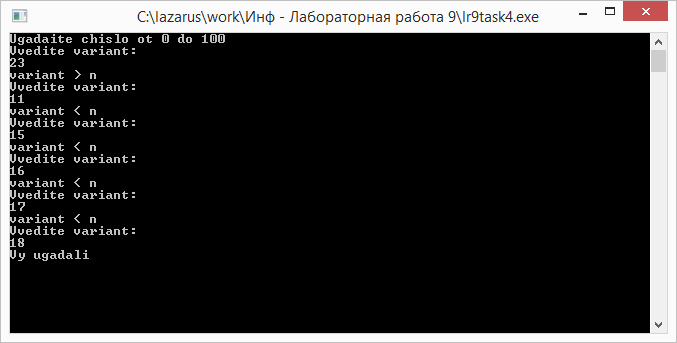
if variant = n then writeln('Vy ugadali');

until variant = n;

readln;

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

Машина задает случайное число. Потом мы вводим своё число, стараясь угадать число машины. Программа даёт нам подсказку – введённое нами число больше или меньше заданного числа. Когда мы угадаем, программа оповестит нас об этом.

1. Вывод: программа работает корректно.