**Лабораторная работа №9**

**Разветвляющиеся вычислительные процессы.**

**Оператор выбора.**

**Цель:** научиться реализовывать алгоритмы разветвляющихся вычислительных процессов и оператора выбора средствами free Pascal.

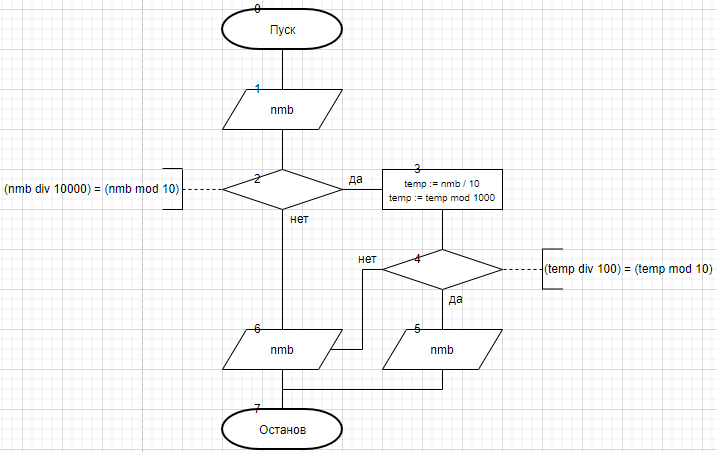
**Используемое оборудование:** пк, PascalABC.

**Задание 1:**

**4.1)** Программа считывает с клавиатуры пятизначное число и определяет, является ли оно палиндромом. (Палиндром – это слово или число, которое одинаково читается как слева направо, так и справа налево, например: 12321,  57975).

**5.1)** nmb div 10000, nmb mod 10, temp = nmb / 10, temp = temp mod 1000, temp div 100, temp mod 10

**6.1)**



**7.1)**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| **nmb** | **Переменная, вводимое пятизначное число** | **int** |
| **temp** | **Переменная, содержит измененное значение nmb** | **int** |

**8.1)**

**program** lab9;

**var**

nmb, temp : integer;

**Begin**

write( 'Введите пятизначное число : ' );

readln( nmb );

**if** ((nmb **div** 10000) = (nmb **mod** 10)) **then**

**begin**

temp := nmb **div** 10;

temp := temp **mod** 1000;

**if** ((temp **div** 100) = (temp **mod** 10)) **then**

writeln( 'Число ', nmb, '- палиндром');

**end**

**else**

writeln( 'Число ', nmb, ' - не палиндром');

**End**.

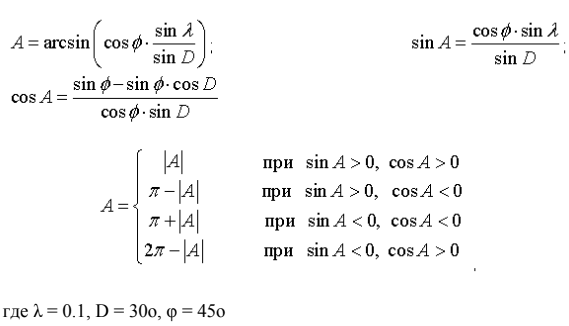
**9.1)**

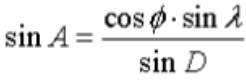
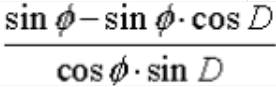


**10.1)** Для решения поставленной задачи мы воспользовались логической конструкцией if then. С начала работы программы запрашивается значение переменной “nmb”, после чего сразу проверяется условие (nmb div 10000) = (nmb mod 10), при истинности которого в переменной “temp”записывается значение переменной “nmb” без первой и последней цифры. Следом проверяется еще одно условие (temp div 100) = (temp mod 10), если истинна, то выводится на экран сообщение о том, что число соответствует всем условиям и является палиндромом, при ложности одно из условий программа выводит на экран об несоответствии числа условиям и программа завершает работу.

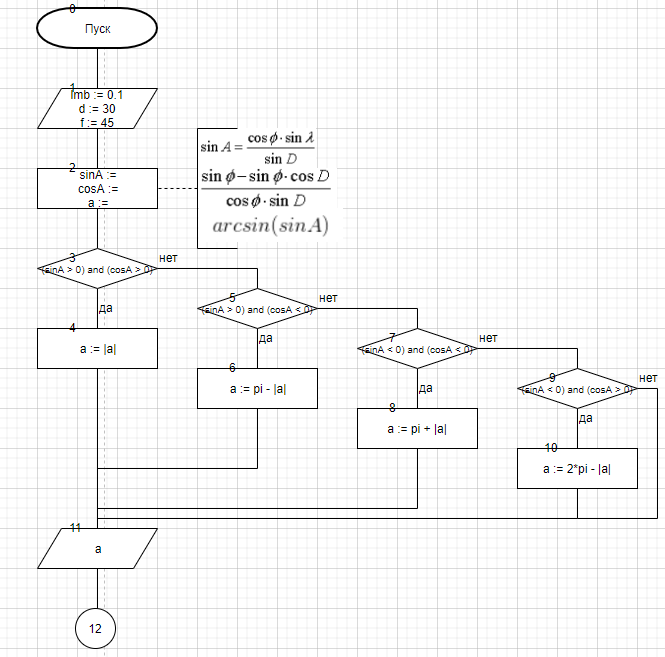
**Задание 2:**

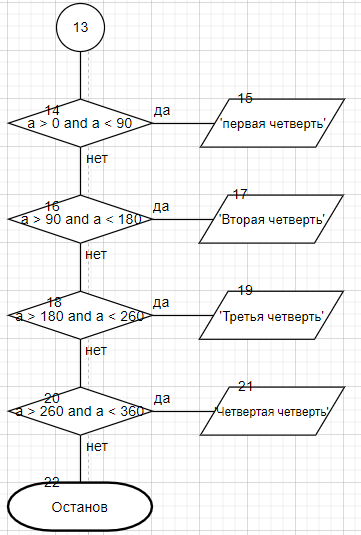
**4.2)** Определить четверть угла азимута А судна, который вычисляется по формулам:



**5.2)** ,,

**6.2)**





**7.2)**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| **d** | **Переменная, угол** | **Int** |
| **f** | **Переменная, угол** | **Int** |
| **lmb** | **Переменная, угол** | **Real** |
| **sinA** | **Переменная, синус угла A** | **Real** |
| **cosA** | **Переменная, косинус угла A** | **Real** |
| **a** | **Переменная, арксинус sinA** | **real** |

**8.2)**

**program** lab9;

**var**

d, f : integer;

lmb, sinA, cosA, a : real;

**Begin**

lmb := 0.1;

d := 30;

f := 45;

sinA := cos(f)\*sin(lmb) / sin(d);

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

a := arcsin(sinA);

**if** ( (sinA > 0) **and** (cosA > 0) ) **then**

a := abs(a)

**else if** ( (sinA > 0) **and** (cosA < 0) ) **then**

a := pi - abs(a)

**else if** ( (sinA < 0) **and** (cosA < 0) ) **then**

a := pi + abs(a)

**else if** ( (sinA < 0) **and** (cosA > 0) ) **then**

a := 2\*pi - abs(a);

write( 'Угол A : ', a:2:5);

**if** (a > 0) **and** (a < 90) **then**

writeln(' - первая четверть');

**if** (a > 90) **and** (a < 180) **then**

writeln(' - вторая четверть');

**if** (a > 180) **and** (a < 260) **then**

writeln('- третья четверть');

**if** (a > 260) **and** (a < 380) **then**

writeln('- четвертая четверть');

**End**.

**9.2)**



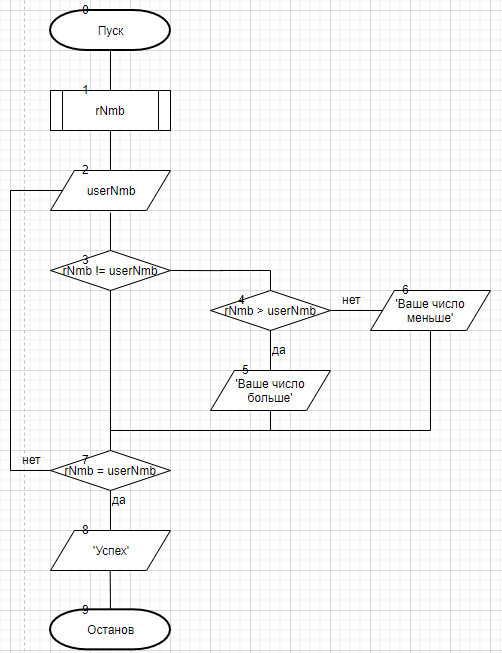
**10.2)** Для решения поставленной задачи мы воспользовались тригонометрическими формулами, а также логическими конструкциями if then, else if else. С начала работы программы выполняется инициализация переменных: lmb, d, f, после чего происходят вычисление переменных: sinA, cosA, a. Дальше начинается проверка условий: (sinA > 0) and (cosA > 0), (sinA > 0) and (cosA < 0), (sinA < 0) and (cosA < 0), (sinA < 0) and (cosA > 0), которые определяют значение a. После проверки условий, на экран выводится значение переменной “a” с точностью до пяти знаков, а дальше с помощью условий: (a > 0) and (a < 90), (a > 90) and (a < 180), (a > 180) and (a < 260), (a > 260) and (a < 380), определяется, к какой четверти принадлежит угол a.

**Задание 3:**

**4.3)** «Угадай число». 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.

**5.3)** -

**6.3)**



**7.3)**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| **userNmb** | **Переменная, вводимое число пользователем** | **int** |
| **rNmb** | **Переменная, случайное число** | **int** |

**8.3)**

**program** lab9;

**uses** crt;

**var**

rNmb, userNmb : integer;

**Begin**

Randomize;

rNmb := random(101);

**repeat**

**begin**

write ('Введите ваше число : ');

readln ( userNmb );

**if** ( userNmb <> rNmb ) **then**

**begin**

**if** ( rNmb > userNmb ) **then**

writeln ('Ваше число меньше')

**else**

writeln ('Ваше число больше');

**end**;

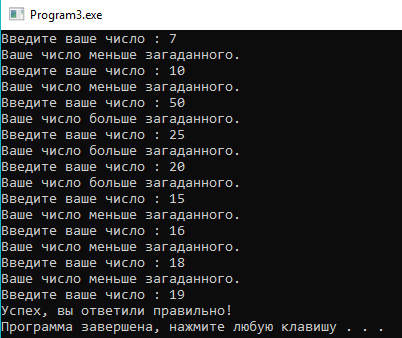
**end**

**until** (rNmb = userNmb);

writeln('Успех, вы ответили правильно!');

**End**.

**9.3)**



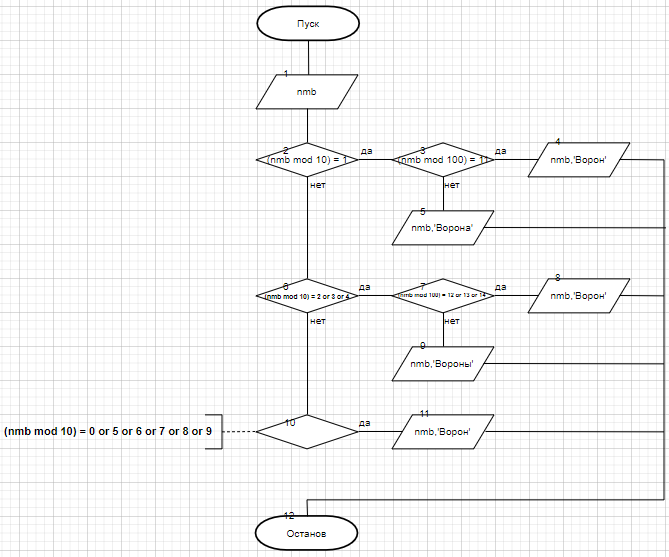
**10.3)** Чтобы решить поставленную задачу, нам помог, цикл конструкции repeat until, а также логическая конструкция if then else. В начале работы программы присваивается случайное значение переменной “rNmb”, после чего начинается цикл с постусловием, где первым действием у пользователя запрашивается число, значение которого записывается в переменную “userNmb”. Дальше в логическом блоке проверяется условие rNmb <> userNmb, при истинности которого проверяется следующее условие на определение величины пользовательского числа, чтобы сообщить больше-ли оно или меньше загаданного числа. В блоке условия цикла проверяется условие rNmb = userNmb, если оно принимает истинность, то программа выводит поздравления и завершается, иначе продолжается работа цикла.

**Задание 4:**

**4.4)** Сформировать вывод слова «ворона» в зависимости от любого числительного, которое вводится с клавиатуры. Например: 1 – ворона, 3 – вороны,  5 – ворон.

**5.4)**nmb mod 10, nmb mod 100

**6.4)**



**7.4)**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| nmb | Переменная, число воронов | Int |

**8.4)**

**program** lab9;

**var**

nmb : integer;

**Begin**

write ('Введите кол-во воронов: ');

readln (nmb);

**case** nmb **mod** 10 **of**

1: **case** nmb **mod** 100 **of**

11:

writeln( nmb, ' - ворон' );

**else**

writeln( nmb,' - ворона' )

**end**;

2,3,4: **case** nmb **mod** 100 **of**

12,13,14:

writeln( nmb, ' - ворон' );

**else**

writeln( nmb,' - вороны' )

**end**;

0,5,6,7,8,9:

writeln( nmb, ' - ворон' );

**end**;

**End**.

**9.4)** , , 

**10.4)**Для решения поставленной задачи был использован цикл конструкции case. В начале работы программы запрашивается ввод числительного, значение которого записывается в переменную “nmb”. Дальше программа попадает в цикл case, где к каждому диапазону чисел соответствуют слова: ворон, ворона, вороны. При помощи операций над значением “nmb” проверяются две последние цифры или только последняя цифра. Когда определяется, к какому диапазону принадлежит nmb, то выводится соответствующее слово. После выхода из цикла, завершается работа программы.

**11)** Мы научились реализовывать алгоритмы разветвляющихся вычислительных процессов и оператора выбора средствами free Pascal.