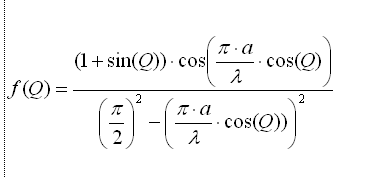
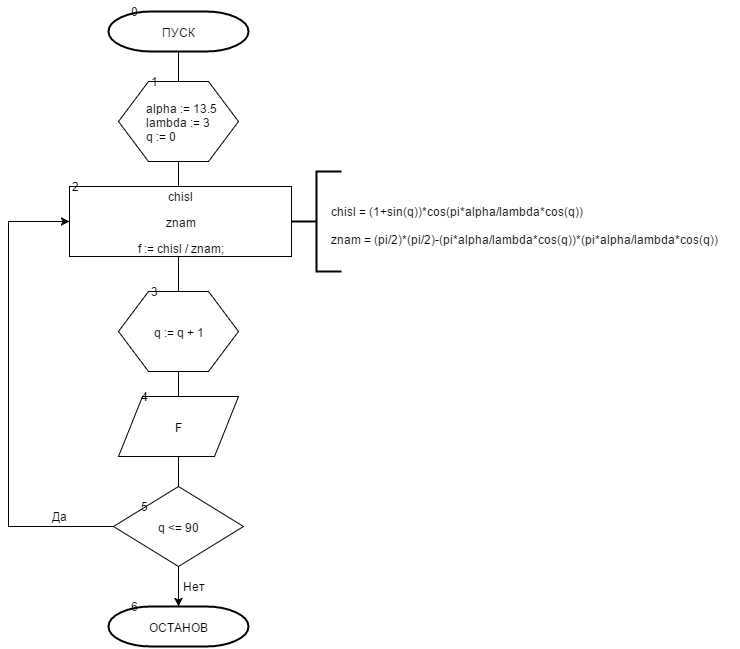
Лабораторная работа №2.

Задание 1.

1. Тема: детерминированные циклические вычислительные процессы с управлением по аргументу.
2. Цель: научиться реализовывать алгоритмы детерминированных циклических вычислительных процессов с управлением по аргументу средствами Free Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: Написать программу, которая позволит рассчитать значения для построения диаграммы направленности антенны в вертикальной плоскости. Q меняется в диапазоне от 0 до 90 градусов с шагом 1 градусов,  *a*= 13.5, лямбда = 3 см.
5. Математическая модель:



1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| q | integer | Градусы наклона |
| f | real | Искомое значение |
| labmda | real | Значение лямбда |
| alpha | real | Значение альфа |
| chisl | real | числитель |
| znam | real | знаменатель |

1. Код программы: program zadanie1;

var

q : integer;

f, alpha, lambda, chisl, znam : real;

begin

alpha := 13.5;

lambda := 3;

for q := 0 to 90 do begin

chisl := (1+sin(q))\*cos(pi\*alpha/lambda\*cos(q));

znam := (pi/2)\*(pi/2)-(pi\*alpha/lambda\*cos(q))\*(pi\*alpha/lambda\*cos(q));

f := chisl/znam;

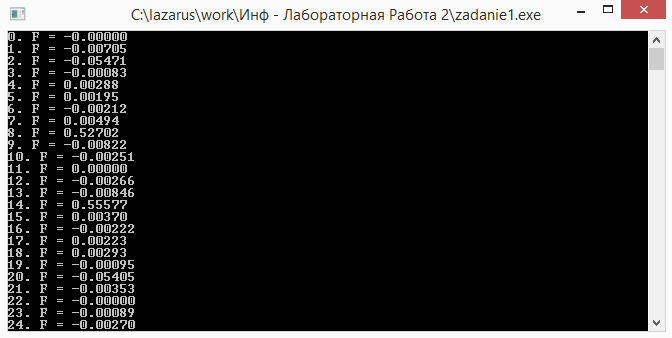
writeln(q,'. F = ',f:3:5);

end;

readln();

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

Наша программа выводит на экран значения для построения диаграммы направленности антенны в вертикальной плоскости в соответствии с данной формулой. Работает она следующим образом: переменным альфа и лямбда присваиваются нужные значения. Затем задается цикл с предусловием и шагом 1. В теле цикла каждый раз происходит вычисление по формуле и вывод результат в удобной для пользователя форме (относительно).

1. Вывод:

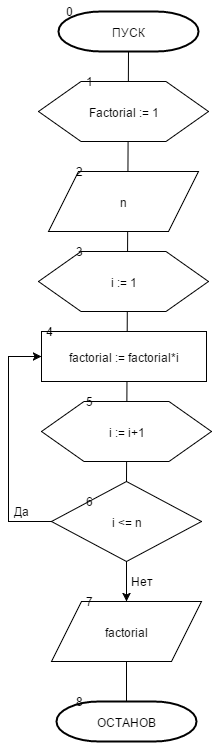
Мы написали программу для вычисления значений для построения диаграммы направленности антенны в вертикальной плоскости.

Задание 2.

1. Тема: детерминированные циклические вычислительные процессы с управлением по аргументу.
2. Цель: научиться реализовывать алгоритмы детерминированных циклических вычислительных процессов с управлением по аргументу средствами Free Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: Написать программу, которая вычисляет факториал числа n, причем n вводится с клавиатуры.
5. Математическая модель:

Factorial = n!

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| q | integer | Градусы наклона |
| f | real | Искомое значение |
| labmda | real | Значение лямбда |
| alpha | real | Значение альфа |
| chisl | real | числитель |
| znam | real | знаменатель |

1. Код программы: program zadanie2;

var

i, n : longint;

factorial : real;

begin

factorial := 1;

writeln('Vvedite N:' ); readln(n);

for i := 1 to n do

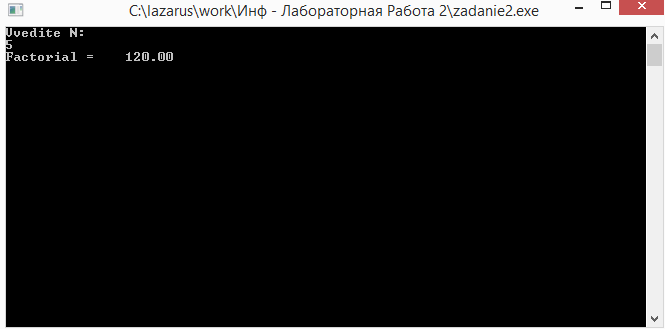
factorial := factorial\*i;

writeln('Factorial = ',factorial:9:2);

readln();

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

Наша программа выводит на экран значения факториала числа n, вводимого с клавиатуры. Сначала присваиваем переменной Factorial значение 1. Во-первых, для вычисления произведения. Во-вторых, для верного вычисления факториала 0, который не входит в цикл. Затем в цикле считаем факториал.

1. Вывод:

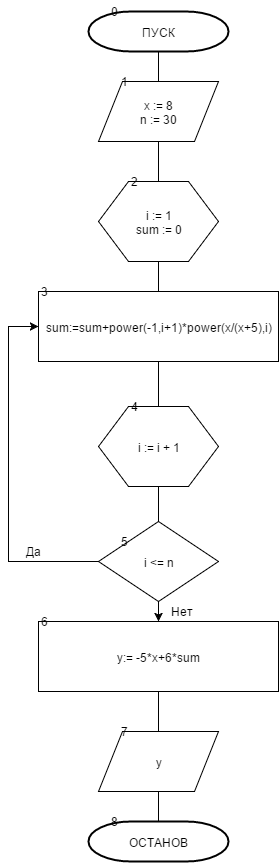
Мы написали программу для вычисления факториала числа n, которое вводится пользователем с клавиатуры.

Задание 3.

1. Тема: детерминированные циклические вычислительные процессы с управлением по аргументу.
2. Цель: научиться реализовывать алгоритмы детерминированных циклических вычислительных процессов с управлением по аргументу средствами Free Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: Написать программу, которая вычисляет значение выражения с заданными значениями x и n.
5. Математическая модель:

Image809

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| x | const | Заданные значения |
| n | const | Заданные значения |
| y | real | Искомое значение |
| sum | real | сумма |
| i | integer | Параметр цикла |

1. Код программы: program zadanie3;

uses math;

const

x = 8;

n = 30;

var

y,sum : real;

i : integer;

begin

sum := 0;

for i:= 1 to n do begin

sum:=sum+power(-1,i+1)\*power(x/(x+5),i);

end;

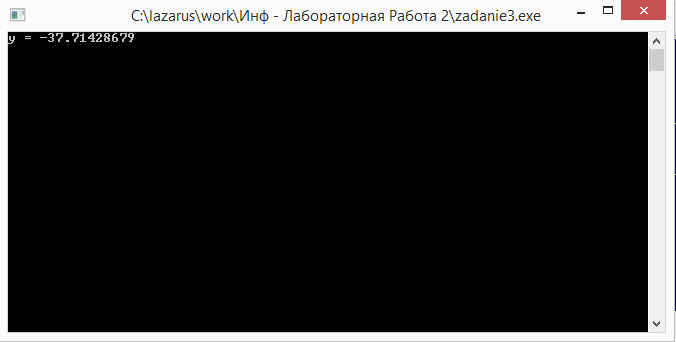
y := -5\*x+6\*sum;

writeln('y = ',y:2:8);

readln();

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

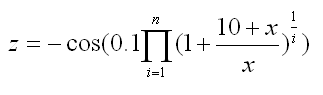
Мы присваиваем переменной sum значение 0 и вычисляем сумму в цикле, используя, в том числе, команду возведения в степень power. Затем, подставив в заданную формулу полученные значения, найдем переменную y и выведем её значение на экран.

1. Вывод:

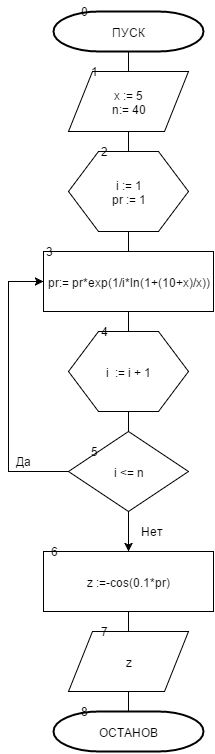
Мы написали программу для вычисления значения выражения с заданными X и N.

Задание 4.

1. Тема: детерминированные циклические вычислительные процессы с управлением по аргументу.
2. Цель: научиться реализовывать алгоритмы детерминированных циклических вычислительных процессов с управлением по аргументу средствами Free Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: Написать программу, которая вычисляет значение выражения с заданными значениями x и n.
5. Математическая модель:



1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| x | const | Заданные значения |
| n | const | Заданные значения |
| z | real | Искомое значение |
| pr | real | произведение |
| i | integer | Параметр цикла |

1. Код программы: program zadanie4;

const

x = 5;

n = 40;

var

z,pr : real;

i : integer;

begin

pr := 1;

for i:= 1 to n do begin

pr := pr\*exp(1/i\*ln(1+(10+x)/x));

end;

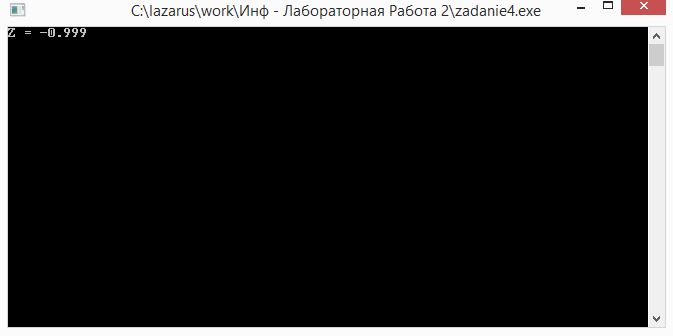
z := -cos(0.1\*pr);

writeln('Z = ',z:2:3);

readln();

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

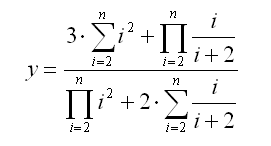
Наша программа выводит на экран значения выражения для указанных Х и N. Сначала она присваивает переменной pr значение 1. Потом начинается цикл: от 1 до n … Считаем произведение. После выхода из цикла, подставляем его в выражение и выводим ответ.

1. Вывод:

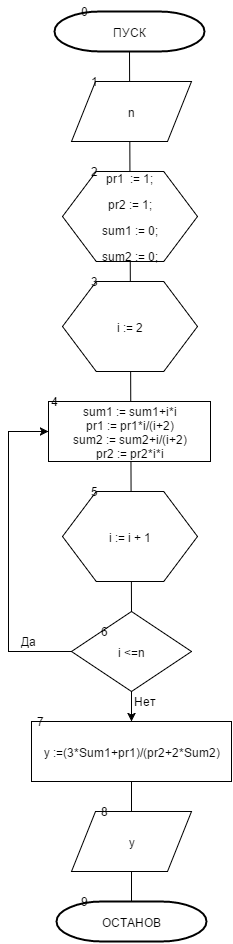
Мы написали программу для вычисления значения выражения с заданными X и N.

Задание 5.

1. Тема: детерминированные циклические вычислительные процессы с управлением по аргументу.
2. Цель: научиться реализовывать алгоритмы детерминированных циклических вычислительных процессов с управлением по аргументу средствами Free Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: Написать программу, которая вычисляет значение выражения. N вводится пользователем.
5. Математическая модель:



1. Блок-схема: См. на след. Стр.



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| y | real | Искомое значение |
| Pr1 | real | произведение |
| Pr2 | real | произведение |
| Sum1 | real | Сумма |
| Sum2 | real | Сумма |
| i | integer | Параметр цикла |
| n | integer | Входные данные |

1. Код программы: program zadanie5;

var

y,pr1,pr2,sum1,sum2 : real;

i,n : integer;

begin

writeln('Vvedite N');

readln(n);

pr1 := 1;

pr2 := 1;

sum1 := 0;

sum2 := 0;

for i:= 2 to n do begin

sum1 := sum1+i\*i;

pr1 := pr1\*i/(i+2);

sum2 := sum2+i/(i+2);

pr2 := pr2\*i\*i;

end;

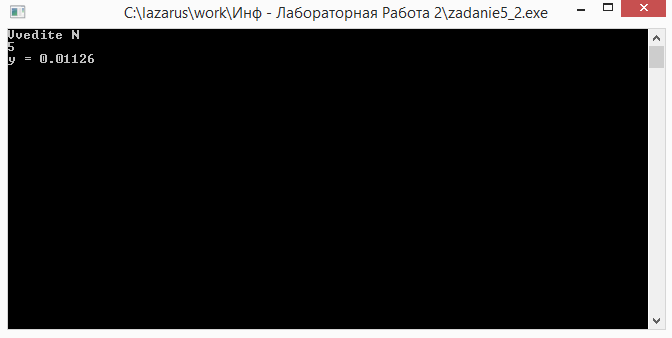
y := (3\*Sum1+pr1)/(pr2+2\*Sum2);

writeln('y = ',y:2:5);

readln();

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

Наша программа выводит получает в качестве входных данных число N. После чего, присвоив соответственно суммам и произведениям значения 0 и 1, начинаются вычисления. После выхода из цикла, подставляем значения в формулу и выводим ответ на экран.

1. Вывод:

Мы написали программу для вычисления значения выражения с n, вводимым пользователем.

Задание 6.

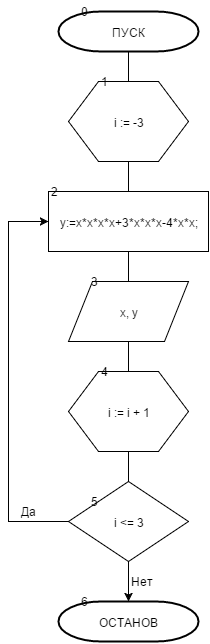
1. Тема: детерминированные циклические вычислительные процессы с управлением по аргументу.
2. Цель: научиться реализовывать алгоритмы детерминированных циклических вычислительных процессов с управлением по аргументу средствами Free Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.

Постановка задачи: Протабулировать функцию на интервале [-3, 3] с шагом 1. Результаты оформить в таблицу.

1. Математическая модель:

****

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| y | real | Искомое значение |
| x | integer | Параметр цикла, аргумент |

1. Код программы: program zadanie6;

var

y : real;

x : integer;

begin

for x:= -3 to 3 do begin

y:= x\*x\*x\*x+3\*x\*x\*x-4\*x\*x;

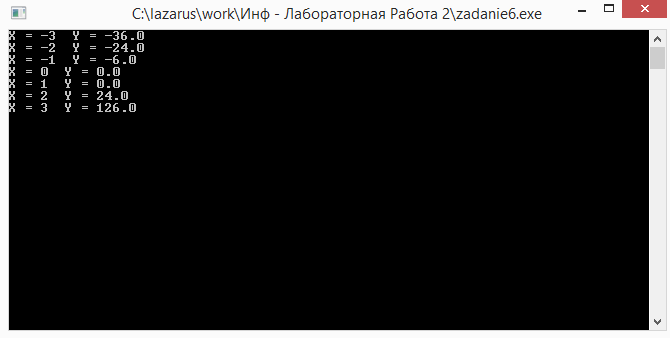
writeln('X = ',x,' ','Y = ',y:2:1);

end;

readln();

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

Наша программа подставляет значения Х в функцию с шагом 1. После этого она выводит значения на экран.

1. Вывод:

Мы написали программу для табулирования функции на заданном интервале с шагом 1.