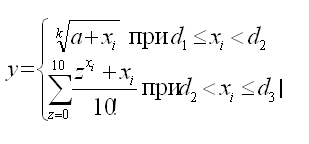
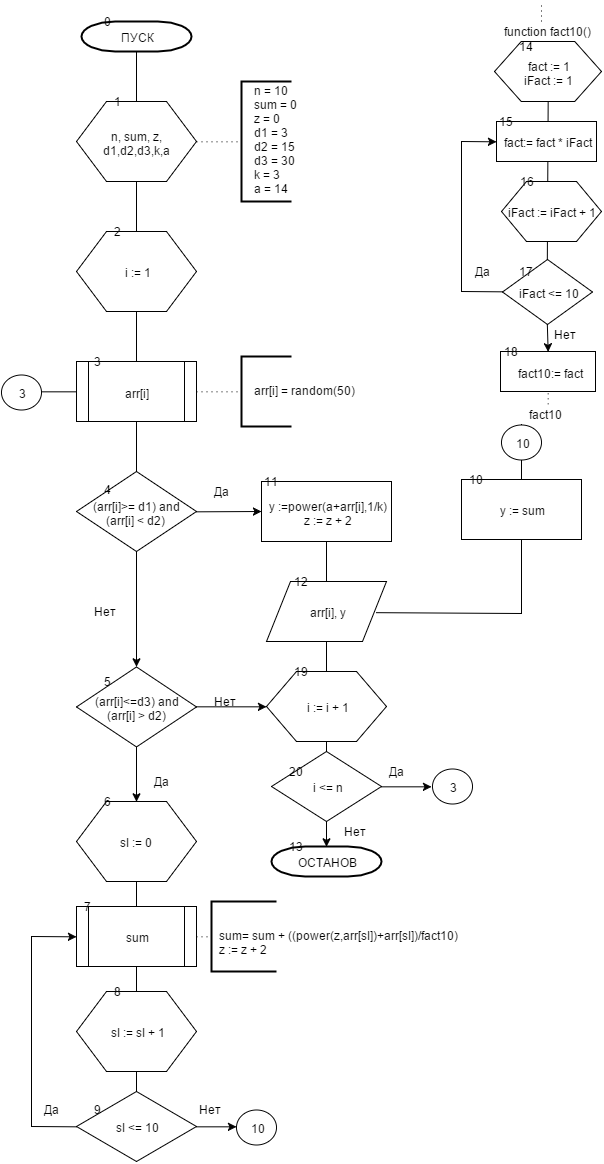
Лабораторная работа №11.

Задание 1.

1. Тема: комбинированные вычислительные процессы.
2. Цель: научиться реализовывать комбинированные вычислительные процессы средствами FreePascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: дан массив чисел X, который состоит из элементов Xi, где i = 1 ÷ n (шаг по Z равен2) . Для элементов массива, попавших в заданный диапазон вычислить y.
5. Математическая модель:



1. Блок-схема: см. ниже



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| i | integer | Параметр цикла, индекс |
| n | integer | Кол-во элементов массива |
| a | integer | Число |
| si | integer | Параметр в цикле нахождения суммы |
| z | integer | Число |
| k | integer | Число |
| arr | Array of integer | Массив со случайными значениями |
| y | real | Искомое число |
| sum | real | Сумма |
| D1 | integer | Для проверки условия |
| D2 | integer | Для проверки условия |
| D3 | integer | Для проверки условия |

1. Код программы:

program lr11\_task1;

uses math;

var

arr : array [1..10] of real;

i,n,a, sI, z, k, d1,d2,d3 : integer;

y,sum : real;

function fact10():integer;

var iFact, fact : integer;

begin

fact := 1;

for iFact := 1 to 10 do

fact := fact \* iFact;

fact10 := fact;

end;

begin

n := 10; // Кол-во элементов массива

sum := 0;

z := 0;

d1 := 3;

d2 := 15;

d3 := 30;

k := 3;

a := 14;

randomize;

for i := 1 to n do begin

arr[i] := random(50);

if (arr[i] >= d1) and (arr[i] < d2) then begin

y := power(a+arr[i],1/k);

writeln('arr[i] = ',arr[i]:2:3,' ','y = ',y:2:3);

z := z + 2;

end;

if (arr[i]<=d3) and (arr[i] > d2) then

begin

sum := 0;

for sI := 0 to 10 do begin

sum := sum + ((power(z,arr[sI])+arr[sI])/fact10);

z := z + 2;

end;

y := sum;

writeln('arr[i] = ',arr[i]:2:2,' y = ',y:2:2);

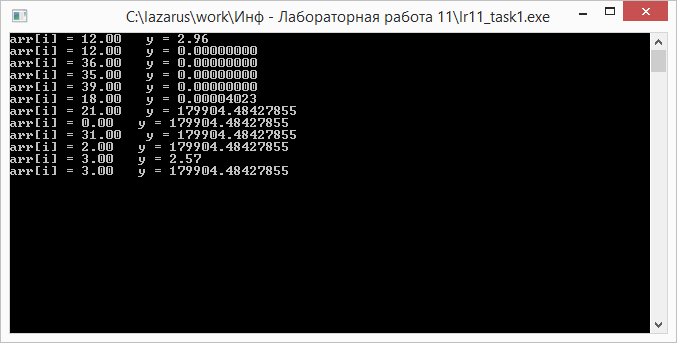
end;

end;

readln;

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

Задаем случайный массив в цикле. Далее проводим выисления в соответствии с поставленными условиями.

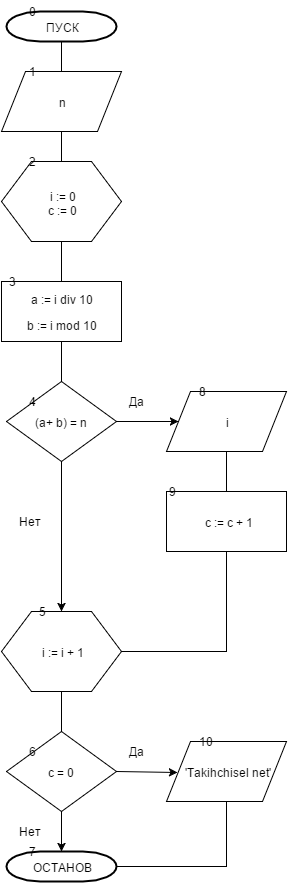
1. Вывод: программа работает корректно.

Задание 2.

1. Тема: комбинированные вычислительные процессы.
2. Цель: научиться реализовывать комбинированные вычислительные процессы средствами FreePascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: программа выдает все числа из диапазона от 0 до 100, сумма цифр которых равна вводимому с клавиатуры числу n (n = 0 ÷ 27).
5. Математическая модель:

Разбиваем все числа от 0 до 100 по разрядам. Если сумма разрядов равна n, выводим разбиваемое число.

1. Блок-схема: см. ниже



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| i | integer | Параметр цикла, число |
| n | integer | Вводимое число |
| a | integer | Десятки |
| b | integer | Единицы |
| c | byte | Проверка условия |

1. Код программы:

program lr11\_task2;

var

n,a,b,i : integer;

c : byte;

begin

writeln('Vvedite chislo 0 <= n <= 27:'); readln(n);

writeln();

c := 0;

for i := 0 to 100 do begin

a := i div 10;

b := i mod 10;

if (a + b) = n then begin

writeln(i);

c := c + 1;

end;

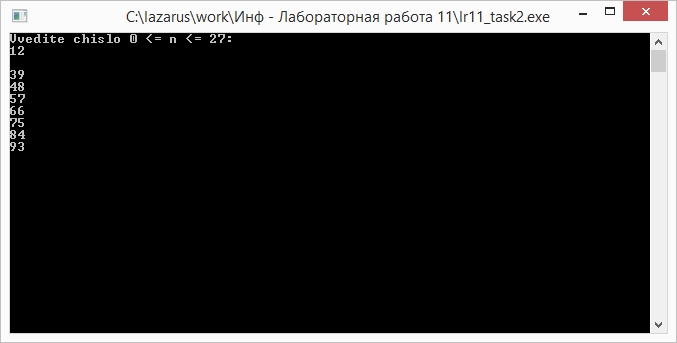
end;

if c = 0 then writeln('Takih chisel net');

readln;

end.

1. Результат работы программы:



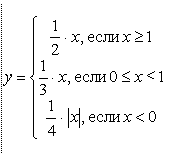
1. Анализ результатов вычисления:

Вводим число n. Разбиваем все числа от 0 до 100 по разрядам. Если сумма разрядов равна n, выводим разбиваемое число. Если проверка не выявила ни одного числа, выводим надпись – таких чисел нет.

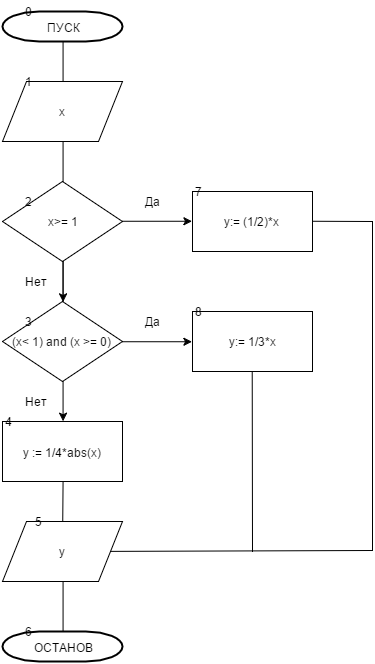
1. Вывод: программа работает корректно.

Задание 3.

1. Тема: комбинированные вычислительные процессы.
2. Цель: научиться реализовывать комбинированные вычислительные процессы средствами FreePascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: вычислить значение функции.
5. Математическая модель:



1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| x | real | Вводимое число |
| y | real | Искомое число |

1. Код программы:

program lr11\_task3;

var

y,x : real;

begin

y := 0;

writeln('Vvedite X: '); readln(x);

if x >= 1 then

y := (1/2)\*x else

if (x < 1) and (x >= 0) then

y := 1/3\*x else

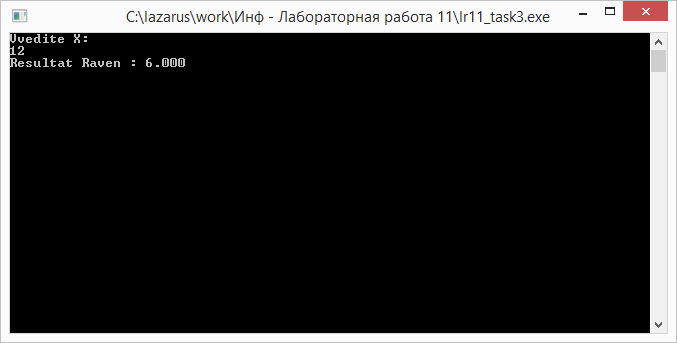
y := 1/4\*abs(x);

writeln('Resultat Raven : ',y:2:3);

readln;

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

С помощью двух проверок проверяем три условия и проводим вычисления.

1. Вывод: программа работает корректно.

Задание 4.

1. Тема: комбинированные вычислительные процессы.
2. Цель: научиться реализовывать комбинированные вычислительные процессы средствами FreePascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: на каком этаже и в каком подъезде 9-этажного дома живет друг, если известен номер его квартиры, и мы знаем, что на каждом этаже располагается 4 квартиры. Номер интересующей нас квартиры вводится с клавиатуры. Вывести номер подъезда и номер этажа, на котором живет друг.
5. Математическая модель:

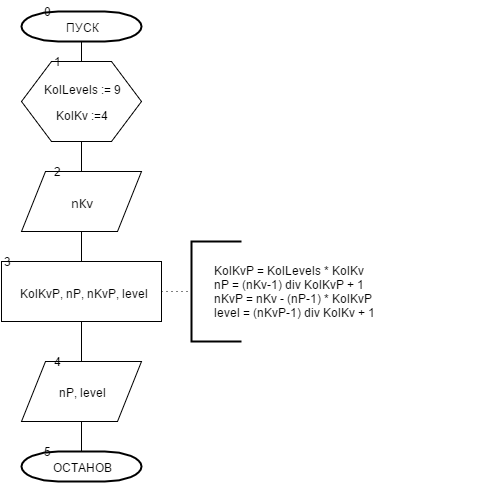
1). 9 \* 4 = 36 – квартир в одном подъезде.

2). Номер квартиры / 36, затем округлить в большую сторону. Так узнаем номер подъезда.

3). Номер квартиры - Номер подъезда \* 36 так мы узнаем, какая по счету квартира друга относительно своего подъезда.

4). Результат действия 3) делим на 4 и округляем в большую сторону – ответ получен.

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| KolLevels | integer | Кол-во этажей в одном подъезде |
| KolKv | integer | Кол-во квартир на одном этаже |
| KolKvP | integer | Кол-во квартир в подъезде |
| nKv | integer | Номер квартиры. Вводимое значение. |
| level | integer | Искомый номер этажа |
| nP | integer | Номер подъезда. Искомое значение. |
| nKvP | integer | Номер квартиры относительно своего подъезда |

1. Код программы:

program lr11\_task4;

var

KolLevels, KolKv, KolKvP, nKv, level,nP, nKvP : integer;

begin

KolLevels := 9;

KolKv := 4;

KolKvP := KolLevels \* KolKv;

writeln('Vvedite nomer kvartiry : '); readln(nKv);

nP := (nKv-1) div KolKvP + 1;

nKvP := nKv - (nP-1) \* KolKvP;

level := (nKvP-1) div KolKv + 1;

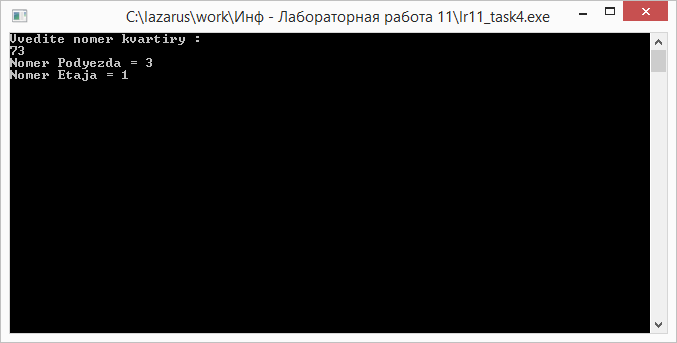
writeln('Nomer Podyezda = ',nP);

writeln('Nomer Etaja = ',level);

readln;

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

Работаем по мат. модели, но с привлечением функций div и mod.

1. Вывод: программа работает корректно.