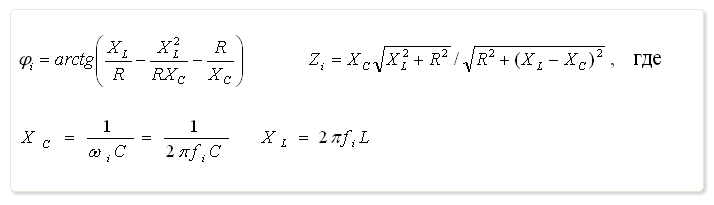
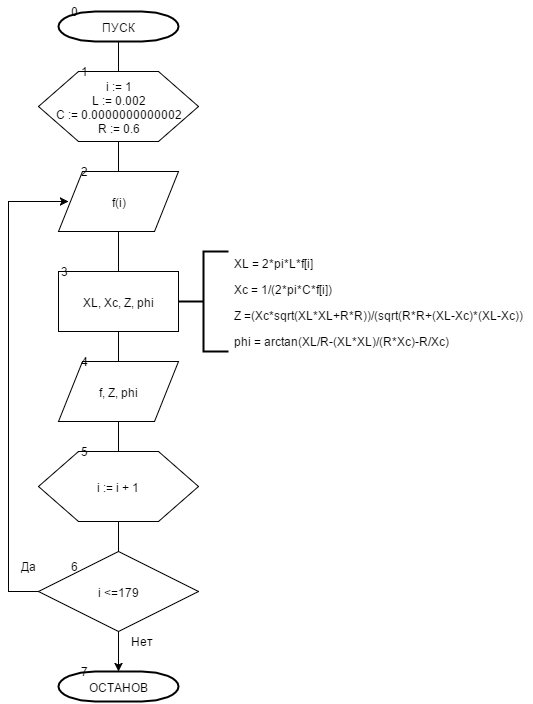
Лабораторная работа №6.

Задание 1.

1. Тема: одномерные массивы.
2. Цель: научиться реализовывать вычисления при помощи массивов на языке FreePascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: Исследовать характер изменения фазового угла phi и реактивного сопротивления колебательного контура Z на различных частотах f (f задано массивом значений).
5. Математическая модель:



1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| XL | real | Индуктивное сопротивление |
| Xc | real | Ёмкостное сопротивление |
| phi | real | Фазовый угол |
| Z | real | Реактивное сопротивление колебательного контура |
| L | real | Индуктивность |
| C | real | Электрическая ёмкость |
| R | real | Сопротивление |
| i | integer | Параметр цикла |
| f | array of integer | Массив, содержащий значения угла наклона |

1. Код программы:

program lr6\_task1\_real\_task1;

var

XL, Xc, phi, Z, L, C, R: real;

i : integer;

f : array[1..179] of integer;

begin

L := 0.002;

C := 0.0000000000002;

R := 0.6;

for i := 1 to 179 do

begin

f[i] := i;

XL := 2\*pi\*L\*f[i];

Xc := 1/(2\*pi\*C\*f[i]);

Z := (Xc\*sqrt(XL\*XL+R\*R))/(sqrt(R\*R+(XL-Xc)\*(XL-Xc)));

phi := arctan(XL/R-(XL\*XL)/(R\*Xc)-R/Xc);

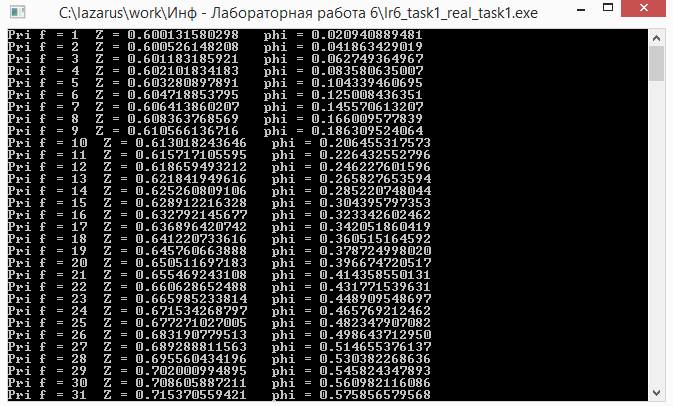
writeln('Pri f = ',f[i],' ','Z = ',z:2:12,' ','phi = ',phi:2:12);

end;

readln;

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

Сначала мы присваиваем нужные значения и заполняем массив в цикле. Потом просто используем конкретные значения массива для вычисления по заданным формулам.

1. Вывод: программа работает корректно.

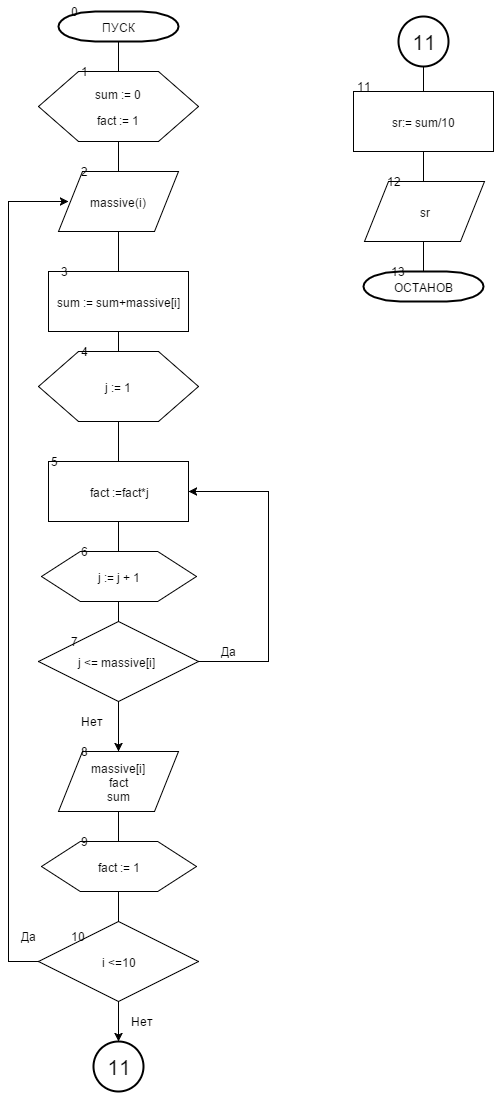
Задание 2.

1. Тема: одномерные массивы.
2. Цель: научиться реализовывать вычисления при помощи массивов на языке FreePascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: Одномерный массив задан случайным образом. Организовать вывод массива. Найти сумму его элементов, вычислить факториал элементов массива, среднее арифметическое его элементов.
5. Математическая модель:

Среднее арифметическое элементов массива равно сумме всех его элементов поделенной на «емкость» массива.

1. Блок-схема:

См. на следующей странице.



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| i | integer | Параметр цикла для заполнения массива |
| j | integer | Параметр цикла для вычисления факториала |
| massive | array of integer | Массив, содержащий 10 значений |

1. Код программы:

program lr6\_task2;

var

i,j : integer;

sum,fact,sr : real;

massive : array[1..10] of integer;

begin

randomize;

sum := 0;

fact := 1;

for i := 1 to 10 do begin

massive[i] := random(i+5);

writeln('massive[',i,'] = ',massive[i]);

sum := sum+massive[i];

for j:=1 to massive[i] do

fact := fact\*j;

writeln('Factorial: ',fact:2:2);

writeln('Summa elementov: ',sum:2:2);

fact := 1;

writeln();

end;

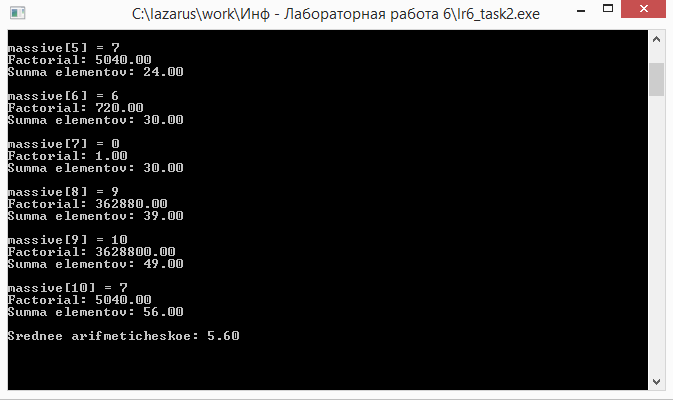
sr := sum/10;

writeln('Srednee arifmeticheskoe: ',sr:2:2);

readln;

end.

1. Результат работы программы:



1. Анализ результатов вычисления: подключив функцию генерирования случайных чисел, мы заполнили наш массив. В процессе заполнения, для каждого элемента мы находим факториал, а также для удобства выводим промежуточную сумму элементов. После нахождения суммы элементов, мы делим её на 10, так как в нашем массиве содержиться 10 значений. Тем самым мы находим среднее арифметическое элементов массива.
2. Вывод: мы написали программу, которая задает одномерный массив случайным образом, выводит его элементы, находит сумму его элементов, вычисляет факториал элементов массива, среднее арифметическое его элементов.

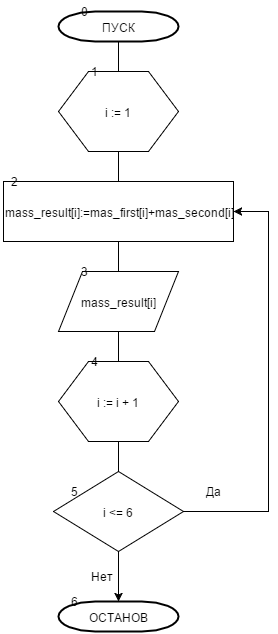
Задание 3.

1. Тема: одномерные массивы.
2. Цель: научиться реализовывать вычисления при помощи массивов на языке FreePascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: Заданы два одномерных массива явно внутри программы. Составить третий одномерный массив, элементы которого являются суммой соответствующих элементов первого и второго массивов.
5. Математическая модель:

Для каждого элемента:

mass\_result[i] = mas\_first[i] + mas\_second[i]

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| i | integer | Параметр цикла для заполнения третьего массива |
| mas\_first | array of integer | Массив, содержащий 6 заданных значений |
| mas\_second | array of integer | Массив, содержащий 6 заданных значений |

1. Код программы:

program lr6\_task3;

const

mas\_first : array[1..6] of integer = (3,5,7,9,2,4);

mas\_second : array[1..6] of integer = (6,10,35,24,13,11);

var

i : integer;

mass\_result : array[1..6] of integer;

begin

for i := 1 to 6 do begin

mass\_result[i] := mas\_first[i]+mas\_second[i];

writeln(mass\_result[i],' = ', mas\_first[i],' + ',mas\_second[i]);

end;

writeln('Proverka');

writeln('mass\_result[1] = ', mass\_result[1]);

writeln('mass\_result[2] = ', mass\_result[2]);

writeln('mass\_result[3] = ', mass\_result[3]);

writeln('mass\_result[4] = ', mass\_result[4]);

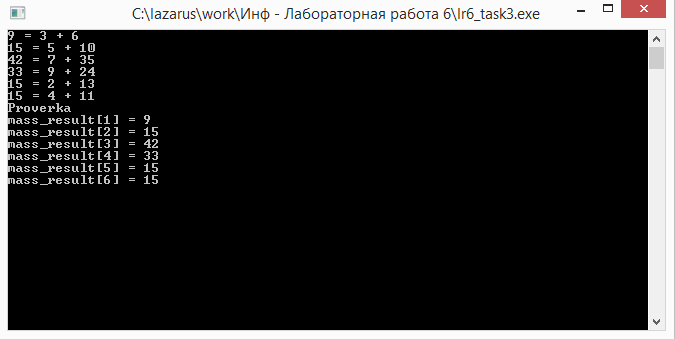
writeln('mass\_result[5] = ', mass\_result[5]);

writeln('mass\_result[6] = ', mass\_result[6]);

readln;

end.

1. Результат работы программы:



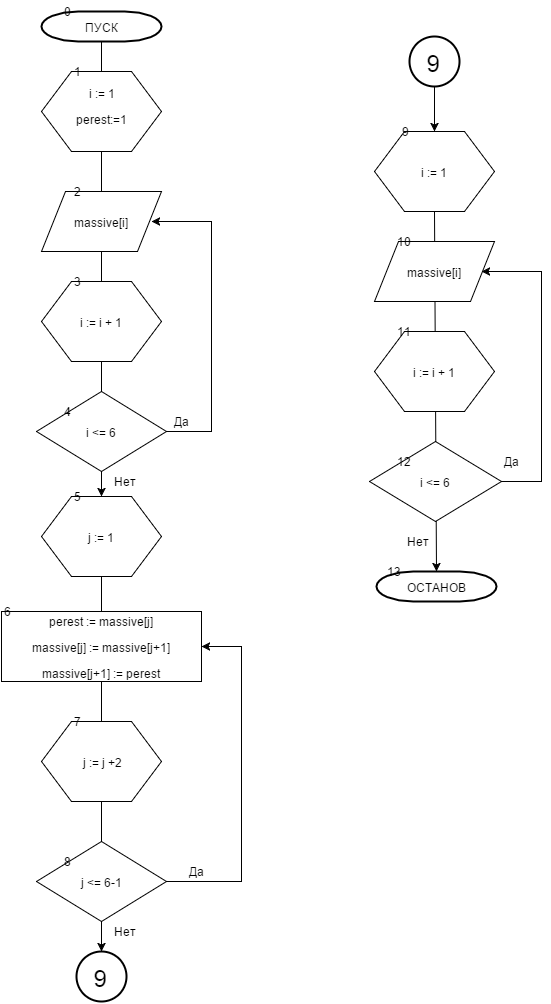
1. Анализ результатов вычисления: задав два одномерных массива явно внутри программы, мы начинаем цикл от 1 до 6 (6 – «емкость» массива) для обработки каждого элемента. Затем присваиваем каждому элементу третьего массива значение, которое равно сумме соответствующих элементов первого и второго массивов. Потом оформляем вывод в форме, удобной для проверки, а затем еще обращаемся к каждому элементу нового массива, чтобы продемонстрировать, что программа работает.
2. Вывод: мы написали программу, которая, имея два заданных одномерных массива, составляет третий одномерный массив, элементы которого являются суммой соответствующих элементов первого и второго массивов.

Задание 4.

1. Тема: одномерные массивы.
2. Цель: научиться реализовывать вычисления при помощи массивов на языке FreePascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: Одномерный массив вводится пользователем с клавиатуры.  Переставить элементы массива, стоящие на четных и нечетных местах. Задачу решить без проверки на четность индексов массива.
5. Математическая модель:

Последовательно обработаем элементы массива с шагом 2. То есть, в нашем случае элементы с индексом 1, 3, 5. Для каждого «индекса» сделаем следующее: присвоим значение обрабатываемого индекса дополнительной переменной. Потом присвоим текущему элементу значение следующего элемента массива, а следующему присвоим значение дополнительной переменной.

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| i | integer | Параметр цикла для заполнения массива и вывода |
| massive | array of integer | Массив, содержащий 6 значений |
| j | integer | Параметр цикла для перестановки |
| perest | integer | Запоминаемое значение для перестановки |

1. Код программы:

program lr6\_task4;

var

massive : array[1..6] of integer;

i,perest,j : integer;

begin

perest:=1;

for i := 1 to 6 do begin

writeln('Vvedite massive[',i,'] :');

readln(massive[i]);

end;

writeln;

j := 1;

while j <= 6-1 do begin

perest := massive[j];

massive[j] := massive[j+1];

massive[j+1] := perest;

j := j + 2;

end;

writeln;

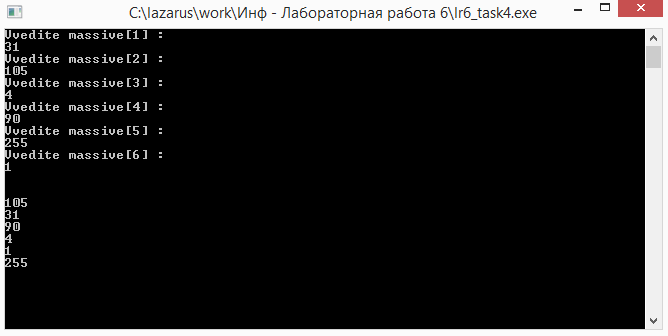
for i := 1 to 6 do

writeln(massive[i]);

readln;

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

Этапы вычисления описаны в мат. модели. Можно добавить, что для ввода элементов массива, их обработки и вывода результата на экран мы используем разные циклы.

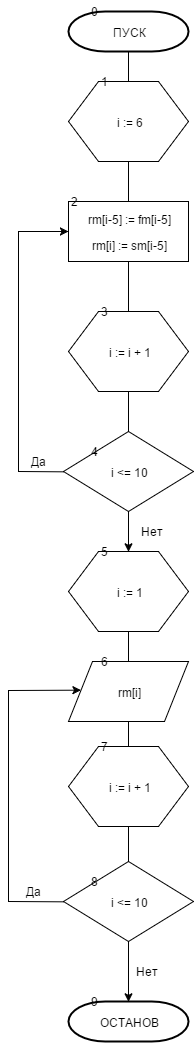
1. Вывод: программа работает корректно.

Задание 5.

1. Тема: одномерные массивы.
2. Цель: научиться реализовывать вычисления при помощи массивов на языке FreePascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: Заданы массивы A(5) и B(5). Получить массив C(10), расположив в начале его элементы массива A, а затем – элементы массива B. Для формирования массива С использовать один цикл.
5. Математическая модель:

Имея два заданных массива, мы должны получить третий, причём его длина будет равна сумме длин данных массивов. Будем присваивать новому массиву значения предыдущих, но с одним условием: первые пять значений из одного цикла, следующие пять – из второго.

1. Блок-схема: см. На следующей странице



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| fm | array of integer | Заданный массив |
| i | integer | Параметр цикла |
| sm | array of integer | Заданный массив |
| rm | array of integer | Искомый массив |

1. Код программы:

program lr6\_task5;

const

fm : array[1..5] of integer = (12,8,15,19,14);

sm : array[1..5] of integer = (48,547,5,1,6);

var

i : integer;

rm : array[1..10] of integer;

begin

for i := 6 to 10 do begin

rm[i-5] := fm[i-5];

rm[i] := sm[i-5];

end;

for i := 1 to 10 do begin

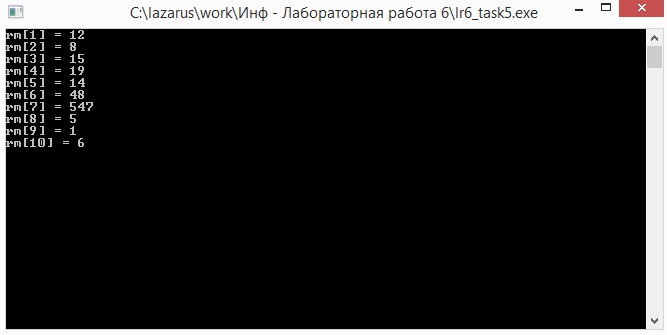
writeln('rm[',i,'] = ',rm[i]);

end;

readln;

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

Начинаем цикл от 6 до 10. Путем манипуляций с индексами заставляем программу сделать то, что нам надо. Хоть цикл и начинается с 6, мы берем элемент из первого массива с индексом [6-5], т.е. [1]. И т.д. до полного заполнения.

1. Вывод: программа работает корректно.