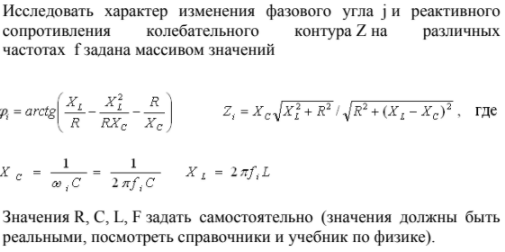
Лабораторная работа №6. Детерминированные ЦВП с управлением по индексу.Одномерные массивы.

Цель л/р: рассмотрение способов реализации алгоритмов ДЦВП с управлением по индексу с использованием одномерных массивов.

Оборудование: ПК, Lazarus, Word, Draw.io.

Задание 1.

1. Математическая модель:



1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| r | Заданное значение, сопротивление | real |
| l | Заданное значение, индуктивность катушки | real |
| c | Заданное значение, ёмкость конденсатора | real |
| freq | Заданная частота, относительно которой считаются остальные частоты, хранящиеся в элементах массива | real |
| f | Массив, предназначенный для хранения значений частот | real |
| i | Параметр цикла for | byte |
| xc | Сопротивление конденсатора | real |
| xl | Индуктивное сопротивление катушки | real |
| len | Длина(количество элементов) массива | byte |
| j | Фазовый угол | real |
| z | Реактивное сопротивление колеательного контура | real |

1. Код программы:

var f:array of real;

i,len:word;

freq,l,c,xc,xl,j,z:real;

r:byte;

begin

freq:=200;

r:=4;

l:=6.37e-4;

c:=1.59e-4;

randomize;

len:=1+random(200);

setlength(f,len);

for i:=0 to len-1 do begin

f[i]:=freq+0.25\*i;

xc:=1/(2\*pi()\*f[i]\*c);

xl:=2\*pi()\*f[i]\*l;

j:=arctan(xl/r-xl\*xl/(r\*xc)-r/xc);

z:=xc\*exp(0.5\*ln(xl\*xl+r\*r))/exp(0.5\*ln(r\*r+(xl-xc)\*(xl-xc)));

writeln('Frequency: ', f[i]:5:2);

writeln('Phase angle: ', j:5:4);

writeln('Impedance: ', z:5:4);

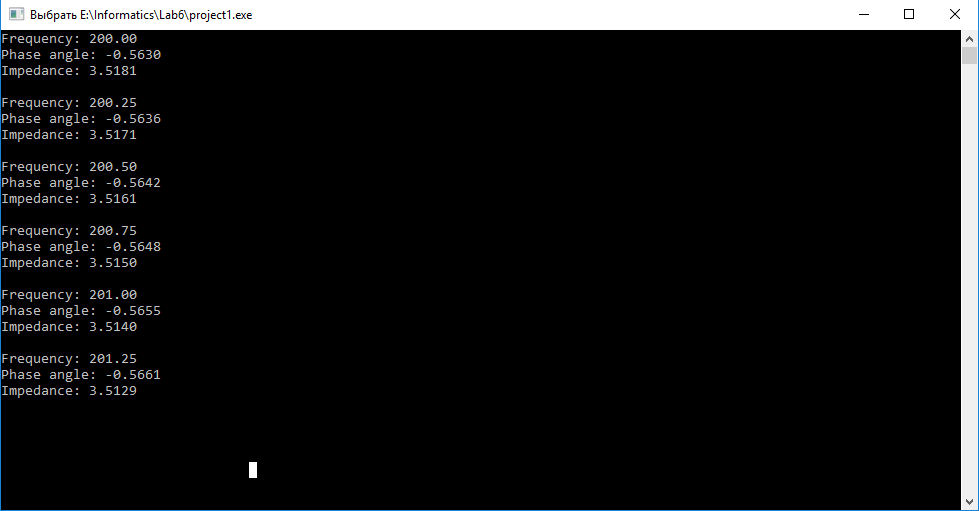
writeln();

end;

readln();

end.

1. Результаты выполненной работы:



1. Анализ результатов вычисления:

Для установления кол-ва элементов массива и вместе с тем итераций цикла for с основными расчётами используется функция random(), в данном случае к ней прибавляется единица, в качестве аргумента взято число 200, итоговое значение длины лежит в диапазоне [1;200]. На каждой итерации первым шагом является подсчёт значения элемента массива, затем на нём базируются основные расчёты. В конце итерации на экран выводятся значения f[i], j, z.

Задание 2.

1. 
2. Математическая модель:
3. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| r | Массив, заданный случайным образом | byte |
| l | Длина(количество элементов) массива | byte |
| i | Парметр цикла for | byte |
| sum | Сумма элементов массива | word |

1. Код программы:

program project2;

var r:array of byte;

l,i:byte;

sum:word;

begin

randomize;

l:=1+random(255);

setlength(r,l);

sum:=0;

for i:=0 to l do begin

r[i]:=random(256);

sum:=sum+r[i];

writeln(r[i]);

end;

writeln('Number of elemnets: ',l);

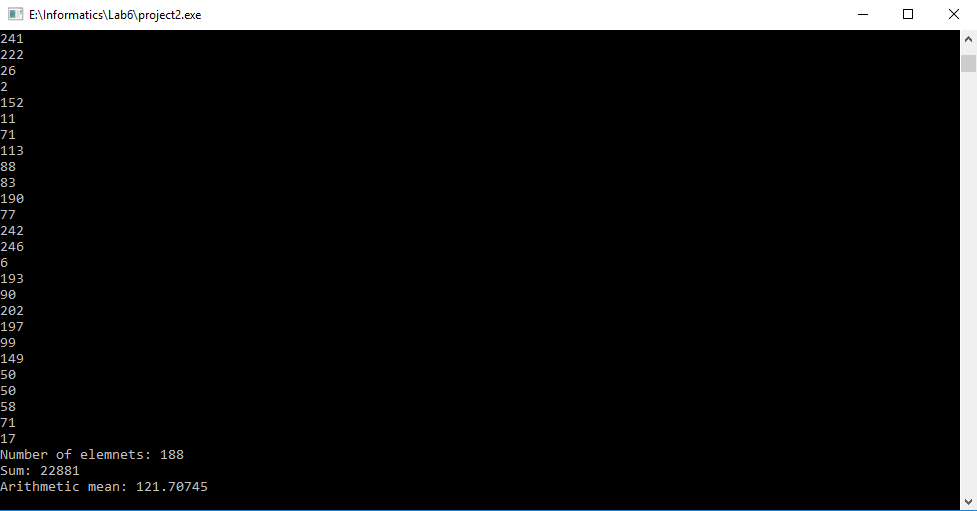
writeln('Sum: ',sum);

writeln('Arithmetic mean: ',sum/l:5:5);

readln();

end.

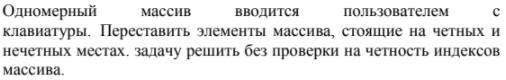
1. Результаты выполненной работы:



1. Анализ результатов вычисления:

Длина массива определяется на начальном этапе программы как 1+random(255). После установления длины массива при помощи функции SetLength обнуляется переменная sum и запускается цикл с параметром i, изменяющимся при выполнении цикла от [0; l-1]. В теле цикла случайное значение от 0 до 255 присваивается элементу с индексом равным значению параметра цикла, затем оно прибавляется к переменной sum. При каждой итерации на экран выводится значение текущего элемента массива. После выполнения цикла выводятся значения длины массива, суммы элементов, и среднее арифметическое его элементов.

Задание 3.



2. Математическая модель:
3. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| f | Массив пользовательских значений | real |
| l | Длина массива | byte |
| c | Переменная для промежуточного хранения f[i] и присвоения его значения f[i+1] | real |
| i | Параметр циклов for и while | byte |

1. Код программы:

program Project3;

var f:array of real;

l,i:byte;

c:real;

begin

randomize;

l:=1+random(15);

setlength(f,l);

for i:=0 to l-1 do begin

readln(f[i]);

end;

writeln('Enter to transform array');

readln();

i:=0;

while i <= l-2 do begin

c:=f[i];

f[i]:=f[i+1];

f[i+1]:=c;

writeln(f[i]:5:3);

writeln(f[i+1]:5:3);

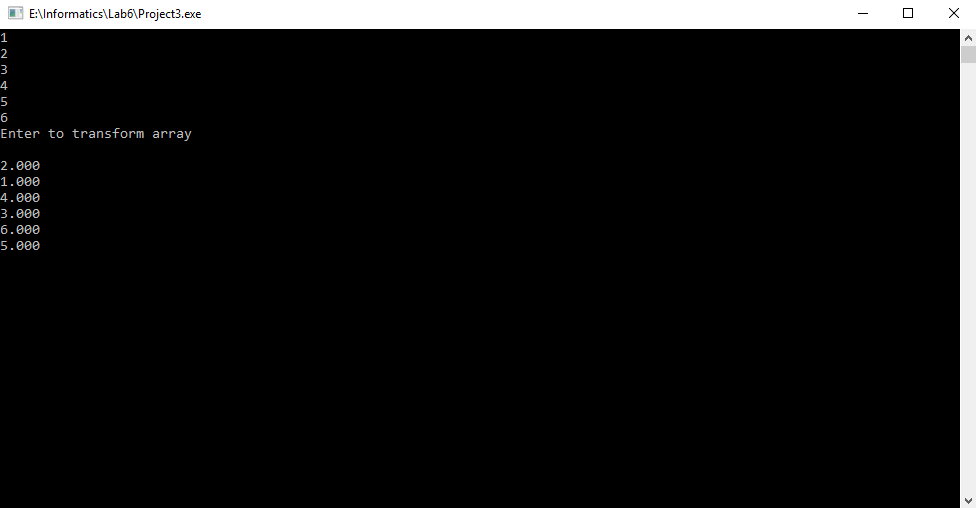
i:=i+2;

end;

readln();

end.

1. Результаты выполненной работы:



1. Анализ результатов вычисления:

При помощи функций random и setlength устанавливается длина массива, после чего используется цикл с параметром i, значение которого изменяется от 0 до l-1. В ходе выполнения цикла пользователь вводит значения элементов массива. При завершении цикла происходит обнуление его параметра i и запускается цикл while, в котором происходит попарная перестановка соседних элементов. При каждой итерации переставленные элементы выводятся на экран.

Задание 4.

1.

2. Математическая модель:

3. Блок-схема:

1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| a | Массив A | integer |
| b | Массив B | integer |
| c | Массив C | integer |
| i | Параметр цикла for | byte |

1. Код программы:

program Project4;

var a:array[0..4] of integer;

b:array[0..4] of integer;

c:array[0..9] of integer;

i:byte;

begin

for i:=0 to 4 do begin

writeln('Enter A element');

readln(a[i]);

end;

for i:=0 to 4 do begin

writeln('Enter B element');

readln(b[i]);

end;

for i:=0 to 4 do begin

c[i]:=a[i];

c[i+5]:=b[i];

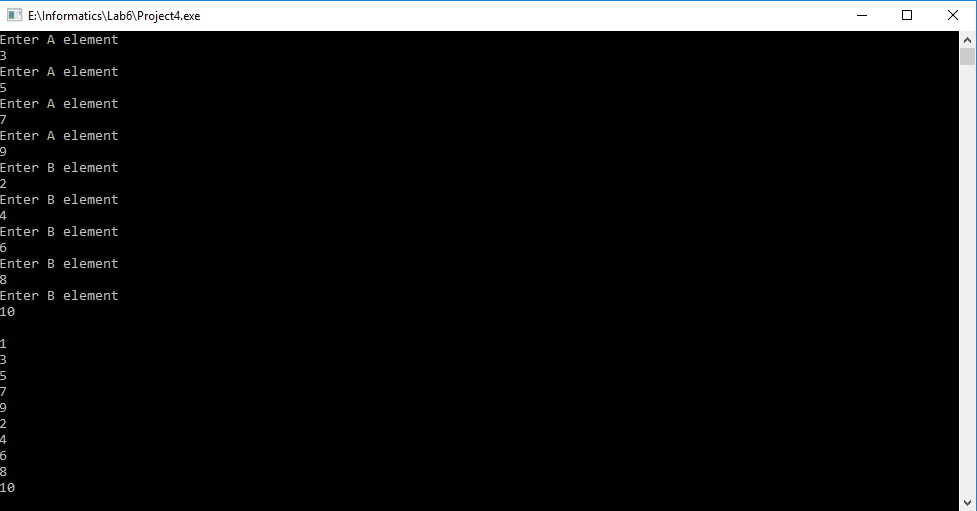
end;

for i:=0 to 9 do

writeln(c[i]);

readln();

end.

1. Результаты выполненной работы:
2. Анализ результатов вычисления:

Программа работает с данным частным случаем. Для работы программы с любыми числами необходимо использование нескольких циклов или оператора условия. В первом цикле программы вводятся элементами массива A, во втором – B, в третьем формируется массив C, в четвёртом осуществляется вывод элементов массива С на экран.

Вывод:

Использование ДЦВП с управлением по индексу позволяет работать с большим кол-вом значений, сохраняя их в памяти, или же обрабатывать уже готовые массивы, используя их элементы для расчётов по заданному алгоритму. Данный вид ДЦВП может быть использован при рассмотрении зависимостей функций от изменения аргумента, а также для изменения расположения элементов в массиве, например, при сортировке.