Лабораторная работа №10. Итерационные ЦВП с управлением по индексу и функции. Вариационный ряд.

Цель л/р: рассмотрение способов организации ИЦВП с управлением по индексу и функции средствами Lazarus и Free Pascal на примере конкретных задач.

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

Задание 1.

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



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| i | Параметр цикла for | byte |
| s | Переменная, предназначенная для хранения суммы чётных чисел | longint |
| a | Идентификатор массива | array of integer |
| l | Длина массива, вводимая пользователем | byte |

1. Код программы:

program Project1;

var a:array of integer;

i,l:byte;

s:longint;

begin

s:=0;

randomize;

writeln('Enter array length(1-255)');

readln(l);

setlength(a,l);

writeln('Generated array:');

i:=0;

while i<=l-1 do begin

a[i]:=1+random(1000);

writeln('a[',i,']: ',a[i]);

if a[i] mod 2 = 0 then

s:=s+a[i];

inc(i);

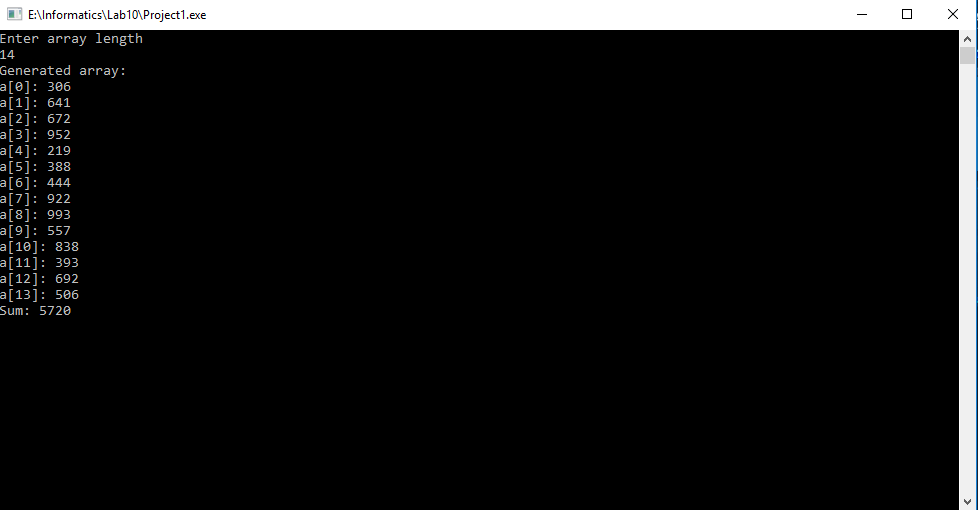
end;

writeln('Sum: ',s);

readln();

end.

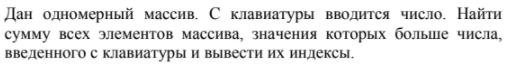
1. Результат выполненной работы:



1. Анализ результатов вычисления:

При запуске программы обнуляется переменная s, в которую будет записываться сумма чётных элементов массива, включается генератор случайных чисел, ожидается ввод значения l – длины массива, далее запускается цикл с начальным значением параметра 0, соответствующим нулевому элементу массива a. В цикле элементу a[i] присваивается случайное значение от 1 до 1000, после чего его индекс и значение выводятся на экран. Далее при помощи оператора условия if проверяется чётность числа, и в случае истинности условия, элемент прибавляется к сумме s. Цикл продолжается до тех пор, пока не будут проверены все элементы массива, т.е. l итераций. При завершении цикла на экран выводится значение суммы чётных элементов s.

Задание 2.

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



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| a | Идентификатор массива | array of byte |
| i | Параметр цикла for | byte |
| x | Вводимое пользователем число для сравнения | integer |
| s | Сумма элементов, удовлетворяющих условию | integer |
| l | Длина массива, вводимая пользователем | byte |

1. Код программы:

program Project2;

var a:array of byte;

i,l:byte;

s,x:integer;

begin

s:=0;

writeln('Enter array length(1-255)');

readln(l);

setlength(a,l);

writeln('Enter the number');

readln(x);

randomize;

writeln('Generated array:');

for i:=0 to l-1 do begin

a[i]:=1+random(255);

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

if a[i]>x then begin

s:=s+a[i];

writeln('a[',i,'] > x');

end;

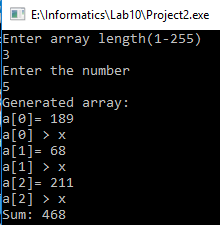
end;

writeln('Sum: ',s);

readln();

end.

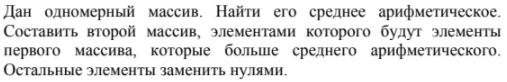
1. Результат выполненной работы:



1. Анализ результатов вычисления:

При запуске программы обнуляется переменная s, предназначенная для суммы чётных удовлетворяющих условию элементов, инициализируется генератор случайных чисел. Затем программа ожидает ввода числа для сравнения x и длины массива l, после чего запускается цикл, в котором элементам массива присваиваются случайные значения в диапазоне [1;255]. Далее в теле цикла осуществляется вывод элемента a[i] и его индекса i, проверяется соответствие условию a[i]>x, в случае его истинности значение текущего элемента a[i] прибавляется к сумме s. Цикл продолжается до тех пор, пока не будут рассмотрены все элементы массива, т.е. всего l итераций. При выходе из цикла на экран выводится итоговое значение суммы.

Задание 3.

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



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| a | Идентификаторов исходного массива | array of real |
| b | Идентификатор полученного массива | array of real |
| i | Параметр циклов for | byte |
| ar\_mean | Среднее арифметическое элементов массива a | real |
| l | Вводимая пользователем длина массивов a и b | byte |

1. Код программы:

program project3;

var a:array of real;

b:array of real;

i,l:byte;

ar\_mean:real;

begin

ar\_mean:=0;

randomize;

writeln('Enter array length(1-255)');

readln(l);

setlength(a,l);

setlength(b,l);

writeln('Generated array:');

for i:=0 to l-1 do begin

a[i]:=1+random(50);

writeln('a[',i,'] = ',a[i]:0:0);

ar\_mean:=ar\_mean+a[i];

end;

ar\_mean:=ar\_mean/(i+1);

writeln('Arithmetical mean = ',ar\_mean:0:3);

readln();

for i:=0 to l-1 do begin

if a[i]>ar\_mean then

b[i]:=a[i]

else

b[i]:=0;

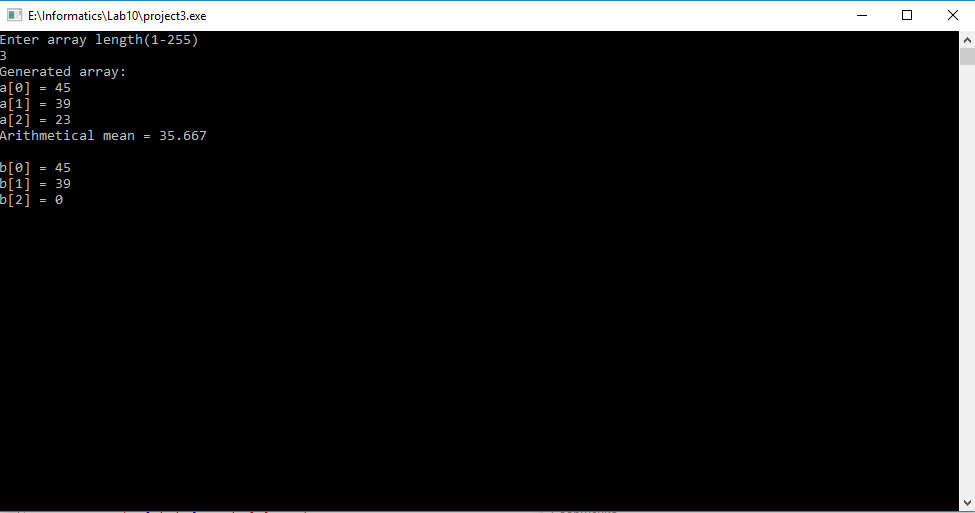
writeln('b[',i,'] = ',b[i]:0:0);

end;

readln();

end.

1. Результат выполненной работы:



1. Анализ результатов вычисления:

При запуске программы обнуляется переменная ar\_mean, предназначенная для хранения значения среднего арифметического элементов массива a, инициализируется генератор случайных чисел. Ожидается ввод пользователем длины l массивов a и b. Далее запускается цикл, в котором элементам массива присваиваются случайные значения в диапазоне [1;50]. Далее в теле цикла осуществляется вывод элемента a[i] и его индекса i, к переменной ar\_mean прибавляется текущий элемент a[i]. Цикл расчитан на l итераций, т.к. массив состоит из l элементов. При выходе из цикла на экран производится расчёт среднего арифметического путём деления ar\_mean на конечное значение параметра цикла i +1, полученное значение выводится на экран. Далее снова обнуляется параметр цикла i, запускается второй цикл, в теле которого происходит проверка условия a[i]>ar\_mean, в случае истинности которого элементу массива b b[i] присваивается значение элемента массива a с тем же индексом b[i]. В противном случае в b[i] записывается 0. После проверки условия на экран выводится полученный элемент. Данный цикл расчитан также на l итераций. При выходе из цикла ожидается пользовательский ввод, программа закрывается.

Задание 4.

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



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| a | Идентификатор массива | array of byte |
| i | Параметр внешнего цикла for | byte |
| j | Параметр внутреннего цикла for | byte |
| p | Переменная для хранения значения a[i] при смене позиций элементов | byte |
| l | Вводимая пользователем длина массива a | byte |

1. Код программы:

program Project4;

var a:array of byte;

i,j,p,l:byte;

begin

randomize;

writeln('Enter array length(1-255)');

readln(l);

setlength(a,l);

writeln('Generated array:');

for i:=0 to l-1 do begin

a[i]:=1+random(255);

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

end;

writeln();

for i:=0 to l-2 do begin

for j:=i+1 to l-1 do begin

if a[i]<a[j] then begin

p:=a[i];

a[i]:=a[j];

a[j]:=p;

end;

end;

end;

writeln('Ordered in descending order array:');

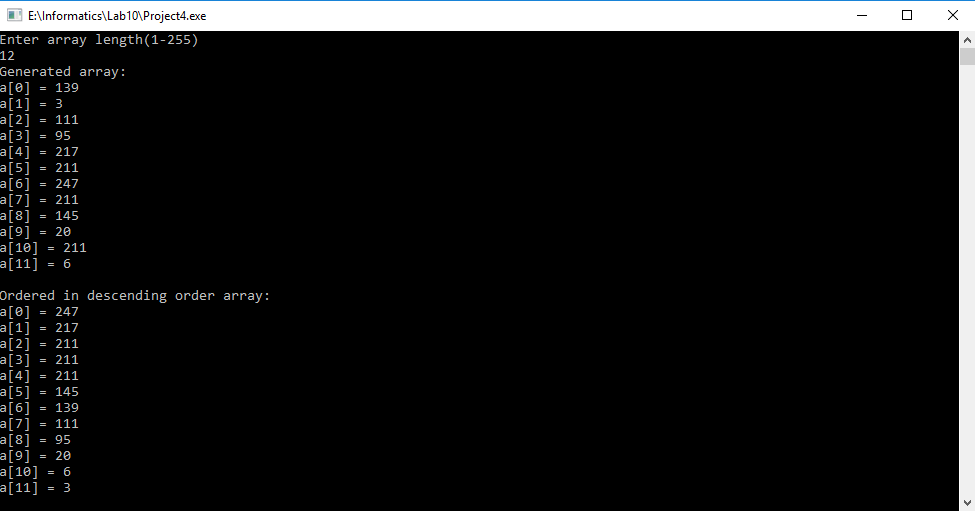
for i:=0 to l-1 do

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

readln();

end.

1. Результат выполненной работы:



1. Анализ результатов вычисления:

При запуске программы инициализируется генератор случайных чисел. После запускается цикл, в котором элементам массива a присваиваются случайные значения в диапазоне [1;255]. Далее в теле цикла осуществляется вывод элемента a[i] и его индекса i. Цикл расчитан на 26 итераций, т.к. массив состоит из 26 элементов. После заполнения массива запускается очередной цикл for с начальным значением параметра i:=0 до 19, внутри которого расположен второй цикл for с параметром j=[i+1;20]. В теле внутреннего цикла for сравниваются элементы массива a[i] и a[j]. Если a[i]<a[j] – элементы меняются местами. Далее на экран выводится элемент a[i] и его индекс i. При завршении данного цикла значение i увеличивается на 1 и снова осуществляется выполнение внутреннего цикла, до тех пор, пока не выполнится 20-я итерация. Таким образом происходит сортировка массива в порядке убывания значений элементов. При завершении цикла снова обнуляется параметр i, запускается последний цикл for, в котором на экран выводятся элементы массива a. После вывода всех элементов ожидается пользовательский ввод, программа закрывается.

Вывод:

Итерационные циклические вычислительные процессы могут использоваться при сортировке элементов массива по определенному заданному признаку. Free Pascal предуматривает организацию данного процесса при помощи циклов for,while,repeat..until. Данный процесс относится к классу полуитерационных, в связи с тем, что выполнение цикла может быть остановлено при достижении истинности выражения в блоке условия.