Альметьевский государственный нефтяной институт

Кафедра информатики

КУРСОВЯ РАБОТА

ПО ДИСЦИПЛИНЕ «ИНФОРМАТИКА»

РАЗДЕЛ: Алгоритмический язык Pascal

НА ТЕМУ:

**«Разработка в среде Turbo Pascal программы сортировки элементов строк матрицы»**

Альметьевск 2010 год

**Решения задачи графическим и программным способами по теме «двумерные массивы»**

Тема курсовой работы: «Разработка в среде Turbo Pascal программы сортировки элементов строк матрицы».

## Постановка задачи

Элементы четных строк матрицы размером n x m упорядочить по возрастанию, элементы нечетных строк- по убыванию. Для отладки программы элементы двумерного массива сформировать с помощью генератора случайных чисел.

Исходными данными являются элементы двумерного массива, которые должны быть созданы по условию задачи с помощью генератора случайных чисел Random. Функция Random без параметра формирует вещественные числа в диапазоне [0,1]. Так как по условию задачи элементами массива должны быть целые числа, то воспользуемся формулой Random(b-a+1)+a, которая будет выдавать случайные целые числа из диапазона [a,b].

В результате решения задачи будет найден четные строки матрицы, элементы которой будут упорядочены по возрастания, и найдены нечетные строки, элементы которой будут упорядочены по убыванию. Для сортировки (упорядочивания) строки массива по возрастанию или убыванию используем алгоритм сортировки обменом.

# Описание алгоритма решения задачи графическим способом

## Укрупненная схема алгоритма

Ввод элементов двухмерного массива

Сортирование элементов

четных строк

по возрастанию, а

нечетных строк

по убыванию

Вывод двухмерного массива

## Детализация укрупненной схемы алгоритма

В программе решается 3 подзадачи:

1. Заполнение двумерного массива;
2. Сортирование элементов четных строк по возрастанию, нечетных строк по убыванию;
3. Вывод преобразованного массива.

## Ввод элементов двумерного массива

Как оговаривалось в постановке задачи, ввод элементов двумерного массива будем осуществлять с помощью генератора случайных чисел. Возьмем, к примеру, интервал от -5 до 15. Тогда, используя формулу Random(b-a+1)+a, получим Random(21)-5. Таким образом, каждый очередной элемент массива будет представлять собой целое число из диапазона [-5, 15] и выводится на экран. Цикл работает до достижения переменной i значения n, то есть до конца массива. Алгоритм заполнения массива соответствующими числами указан ниже:

i = 1, n

j = 1, m

А[i, j]:=Random(21)-5

A[i,j]

## Сортирование элементов четных строк по возрастанию, нечетных строк по убыванию

Сортирование элементов четных строк по возрастанию работает следующим образом: в начале определяем, является ли строка четной. Это определяется с помощью условия If I mod 2 =0, если строка четная, то номер строки делится на 2 без остатка.

В начале находим элемент с наименьшим значением (min) во всей строке и ставится на первое место, а первый элемент при этом ставится на место, где располагался наименьший элемент. Затем наименьший элемент отыскивается уже среди чисел со второго по n элемент, и также меняются местами минимальный среди них и второй элемент и т.д. Для того чтобы не терялись значение элементов массива вместо которых ставятся элементы с наименьшим значение, введена переменная S. Если условие If I mod 2 =0 произошло с остатком, то строка нечетная и начинается сортировка по убыванию. Находим элемент с наименьшим значение во всей строке. Ставим его на последнее место, а последний элемент при этом ставится на место, где располагался наименьший элемент. Чтобы элемент не терялся используем переменную S.

i /2 =0

Min:=A[I,k]

J:=1,m

a[I,j]<min

Min:=a[I,j]

S:=a[I,k]

A[i,k]:=min

A[I,l]:=S

Min:=A[I,k]

i:=1,n

a[I,j]>min

Min:=a[I,j]

L:=j

S:=a[I,k]

A[i,k]:=min

J:=1,m

L:=j

A[I,l]:=S

+

+

+

\_-

-

-

## Вывод преобразованного массива

После сортировки выводим полученный массив на экран. Вывод элементов двумерного массива осуществляется при помощи цикла с параметром.

i = 1, n

j = 1, m

A[i,j]

Блок-схема алгоритма

i = 1, n

j = 1, m

А[i, j]:=Random(21)-5

A[i,j]

i /2 =0

Min:=A[I,k]

J:=1,m

a[I,j]<min

Min:=a[I,j]

S:=a[I,k]

A[i,k]:=min

A[I,l]:=S

Min:=A[I,k]

i:=1,n

a[I,j]>min

Min:=a[I,j]

L:=j

S:=a[I,k]

A[i,k]:=min

J:=1,m

L:=j

A[I,l]:=S

+

+

+

\_-

-

-

i = 1, n

j = 1, m

A[i,j]

# Разработка программы на языке Pascal

Программа начинается со служебного слова Program, после которого следует заголовок программы. В данном случае это massiv. Далее включаем раздел Uses для использования модуля CRT, который применяется для управления работой экрана в текстовом режиме. После названия программы и идентификации используемых модулей следует раздел объявления констант (const) и переменных (var).

В данной программе в разделе констант объявлены константы n=4 (количество строк массива) и m = 5 (количество столбцов массива). В разделе переменных описан целочисленный массив под именем a, целочисленные переменные i, j – счетчики циклов, min – минимальный элемент, imin – индекс минимального элемента, minst – минимальный элемент столбца, k – вспомогательная переменная для сортировки элементов.

Тело программы или раздел операторов начинается со слова begin и заканчивается end. В этом разделе описываем действия, которые должна выполнить программа согласно выбранного алгоритма. Так как в программе подразумевается ввод данных с экрана и вывод полученных результатов на экран, перед началом программы его необходимо очистить от ненужной информации. Это проделывает процедура clrscr, которая описана в модуле Crt.

Перед первым обращением к функции random необходимо с по-мощью вызова процедуры randomize инициализировать программный генератор случайных чисел, иначе при каждом запуске программы датчик будет выдавать одни и те же числа.

## Описание блоков укрупненной схемы алгоритма на языке Pascal

## Ввод элементов двумерного массива

Для того, чтобы прокомментировать, что вначале будет выведен исходный массив на экран, используем оператор writeln ('исходная матрица:').

Рассмотренный фрагмент блок-схемы для реализации ввода элементов двумерного массива на языке Pascal будет представлен в следующем виде:

for i:=1 to n do

begin

for i:=1 to n do begin

a[i,j]:=random(21)-5;

write(a[i,j]); end;

writeln;

end;

writeln;

Следующий далее оператор writeln без параметров просто переводит курсор на другую строку. Это необходимо для того, чтобы данные выводились на экран в виде матрицы. Добавим еще оператор writeln без параметров для более удобного восприятия информации с экрана.

## Сортирование элементов четных строк по возрастанию, нечетных строк по убыванию

Далее начинаем обработку массива. Обработка массива осуществляется в цикле с параметром for i:=1 to n do по строкам массива. Далее с помощью условия определяем if i mod 2=0 then определяем четность строки.

Задаем цикл с параметром for k:=1 to m do, k это переменная для обозначение столбца массива. Определяем первый элемент массива как минимальный, для того чтобы можно было сравнивать с ним другие элементы и отыскать минимальный из них. Условием if a[i,j]<min сравниваются два элемента одной строки. Если условие выполняется то в минимальный присваивается следующий элемент min:=A[i,j]. Затем меняем местами эти элементы. Переменная S используем промежуточный элемент для хранение элемента массива S:=A[i,k]. Потом заносим в массив A[i,k] минимальный элемент A[i,k]:=min, а в массив A[i,l] восстанавливаем значение из переменной S (A[i,l]:=S).

Если первое условие не выполнилось if i mod 2=0 then то происходит сортировка нечетной строки по убыванию.

Обработка массива начинается с цикла с параметром. Задаем переменную max, в эту перемену присваиваем значение первого элемента max:=a[i,k]. Задаем цикл с параметром for j:=k to m do. Проверяем выполнения условия if a[i,j]>max, если условие выполняется то в переменную max заносим элемент массива a[i,j].

В переменную S заносим значение массиваA[i,k], в массив A[i,k] заносим значение максимального элемента max и восстанавливаем в переменную A[i,l]значение переменной S.

for i:=1 to n do begin

if i mod 2=0 then for k:=1 to m do begin

min:=a[i,k];

for j:=k to m do

if a[i,j]<min then

begin min:=A[i,j]; l:=j;

S:=A[i,k];

A[i,k]:=min;

A[i,l]:=S end; end

else for k:=1 to m do begin

max:=a[i,k];

for j:=k to m do

if a[i,j]>max then

begin max:=A[i,j]; l:=j;

S:=A[i,k];

A[i,k]:=max;

A[i,l]:=S end; end;

end;

## Вывод преобразованного массива

После сортировки выводим полученный массив на экран стандартными средствами вывода:

for i:=1 to n do

begin

for i:=1 to n do

write(a[i,j]:4);

writeln;

end;

где write(a[i,j]:4)– вывод элементов двумерного массива в строку

с указанием количества занимаемых позиций

## Листинг программы

Program massiv;

Uses crt;

Const n=5; m=5;

Var a: array[1..n, 1..m] of integer;

i, j, k,min,max, l, s: integer;

Begin

Clrscr; randomize;

Writeln('исходный:'); begin

For i:=1 to n do

Begin

for j:=1 to m do begin

a[i,j]:=random(21)-5; write(a[i,j]:4); end;

writeln; End; Writeln;end;

for i:=1 to n do begin

if i mod 2=0 then for k:=1 to m do begin

min:=a[i,k];

for j:=k to m do

if a[i,j]<min then

begin min:=A[i,j]; l:=j;

S:=A[i,k];

A[i,k]:=min;

A[i,l]:=S end; end

else for k:=1 to m do begin

max:=a[i,k];

for j:=k to m do

if a[i,j]>max then

begin max:=A[i,j]; l:=j;

S:=A[i,k];

A[i,k]:=max;

A[i,l]:=S end; end;

end;

begin

writeln(результат:'); For i:=1 to n do Begin

for j:=1 to n do write(a[i,j]:4);

Writeln;

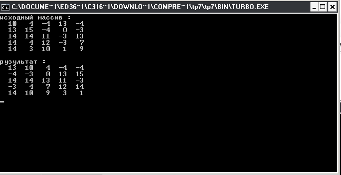
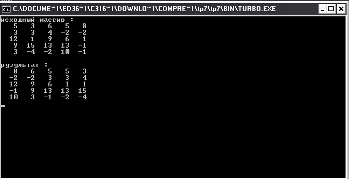
end; end;

readln;

End.

## Тестирование программы

Ниже приведены результаты выполнения программы на примере различных входных данных.



# Список используемой литературы

1. А.И. Гусева. Учимся программировать: Pascal 7.0. М. диалог-мифи, 2000г.

2. А.Ф. Иванов, О.Н. Потапова, Г.Л. Салихова А. Основные алгоритмы языка Pascal, учебное пособие. 2007г.

3. А.Ф. Иванов А. Программирование на алгоритмическом языке Pascal, 2000г.

4. Семакин И.Г., Шестаков А.П. Основы программирования. М.:Академия, 2003г.