1. Лабораторная работа № 5

«ОБРАБОТКА ДВУМЕРНЫХ МАССИВОВ С ПОМОЩЬЮ

ПРОЦЕДУР И ФУНКЦИЙ»

# Цель работы

Изучить основные принципы обработки двумерных массивов, получить навыки разработки программ блочной структуры, исследовать способы передачи параметров в процедуры и функции.

# Постановка задачи

Получить навыки работы с процедурами и функциями, изучить основные принципы обработки двумерных массивов.

# Ход выполнения работы

# Описание алгоритма решения задачи

# Математическое обоснование алгоритма

Матрицей m на n называется прямоугольная таблица чисел, состоящая из m строк и n столбцов. Каждый элемент матрицы A имеет два индекса: i - номер строки и j– номер столбца. Матрица, у которой число строк равно числу столбцов, называется квадратной. Элементы, стоящие по диагонали, идущей из верхнего левого угла, образуют главную диагональ матрицы.

# Определение входных и выходных данных

На вход программы поступают элементы матрицы, записанные строчка за строчкой.

В качестве ответа выводится два целых числа: первое – сумма элементов в тех строках, которые не содержат отрицательных элементов, второе - минимум среди сумм элементов диагоналей, параллельных главной диагонали матрицы.

# Структурная схема алгоритма

Структурная схема программы представлена на Рисунках 5.1 и 5.2.

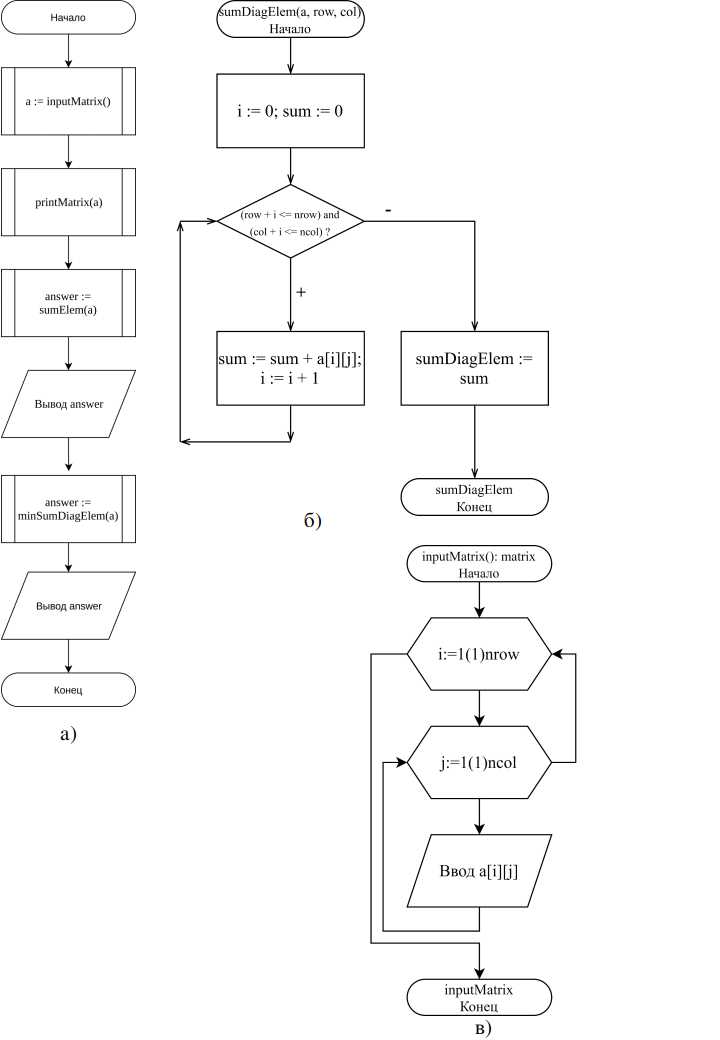


Рисунок 5.1 – Структурные схемы: а) основная программа; б) функция sumDiagElem; в) функция inputMatrix



Рисунок 5.2 – Структурные схемы: а) процедура printMatrix; б) функция sumElem; в) функция minSumDiagElem

# Текст программы

Program LW\_5;

Const nrow = 100; ncol = 100;

Type matrix=array[1..nrow] of array[1..ncol] of integer;

Var a: matrix; answer: integer;

{----------------Функция-вычисления-суммы-элементов-в-строках----------------------}

Function sumElem(const a: matrix): integer;

Var i, j, sumRow, sumAnswer: integer;

begin

sumRow := 0; sumAnswer := 0;

for i := 1 to nrow do

begin

sumRow := 0;

for j := 1 to ncol do

begin

if a[i][j] < 0 then begin sumRow := 0; break end; sumRow := sumRow + a[i][j]

end;

sumAnswer := sumAnswer + sumRow

end;

sumElem := sumAnswer

end;

{------------------------------Функция-вычисления-суммы-элементов---------------------}

{---------------------------------в-диагонали,-параллельной-главной-----------------------}

Function sumDiagElem(const a: matrix; const row: integer; const col: integer): integer;

Var sum: integer; i: integer;

begin

i := 0; sum := 0;

{Магическая функция...}

while ((row + i <= nrow) and (col + i <= ncol)) do

begin sum := sum + a[row + i][col + i]; i := i + 1 end;

sumDiagElem := sum;

end;

{-------------------Функция-вычисления-минимальной-суммы----------------------------}

{------------------------------элементов-по-диагонали-----------------------------------------}

Function minSumDiagElem(const a: matrix): integer;

Var minSum, sum, i: integer;

begin

minSum := sumDiagElem(a, 1, 1);

{Подсчёт сумм, диагонали которых начинаются с первой строчки}

for i := 2 to ncol do

begin

sum := sumDiagElem(a, 1, i);

if sum < minSum then

minSum := sum;

end;

{Подсчёт сумм, диагонали которых начинаются с первого столбца}

for i := 2 to nrow do

begin

sum := sumDiagElem(a, i, 1);

if sum < minSum then

minSum := sum;

end;

minSumDiagElem := minSum

end;

{--------------------------------------Функция-ввода-матрицы---------------------------------}

Function inputMatrix(): matrix;

Var i, j: integer; a: matrix;

begin

for i := 1 to nrow do for j := 1 to ncol do readln(a[i][j]);

inputMatrix := a;

end;

{-----------------------------------Процедура-вывода-матрицы--------------------------------}

Procedure printMatrix(const a: matrix);

Var i, j: integer;

begin

for i := 1 to nrow do

begin

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

end

end;

{-----------------------------------Основная-программа-----------------------------------------}

BEGIN

{Ввод матрицы}

writeln('Введите элементы матрицы ', nrow, 'x', ncol, ':');

a := inputMatrix();

{Контрольный вывод матрицы}

writeln('Введена матрица:'); printMatrix(a);

{Вывод ответов}

writeln('Ответ 1: ', sumElem(a)); writeln('Ответ 2: ', minSumDiagElem(a));

writeln(‘Конец работы программы.’); readln();

END.

# Описание тестовых примеров

Тестовые примеры 1-5 представлены на Рисунках 5.3-5.6.

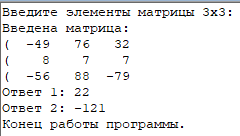


Рисунок 5.3 – Тест программы, если введена квадратная матрица 3-го порядка

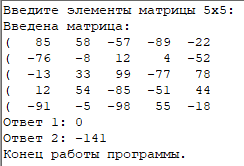


Рисунок 5.4 – Тест программы для квадратной матрицы 5-го порядка

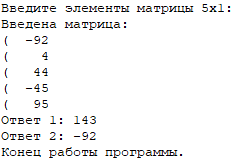


Рисунок 5.5 ­– Тест программы для матрицы-столбца

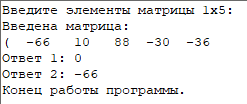


Рисунок 5.6 – Тест программы для матрицы-строки

Из тестовых примеров 1-3 следует, что программа проводит верные вычисления. Из тестовых примеров 4-5 следует, что программа работает корректно для матрицы-столбца и матрицы-строки несмотря на то, что по условию задания вводится квадратная матрица.

# Вывод

В ходе выполнения данной лабораторной работы были получены навыки работы с процедурами и функциями, с двумерными массивами. Выявлено, что при работе с матрицами удобно использовать двумерные массивы; в больших программах более рационально и эстетично использовать функции и процедуры.