Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Основы алгоритмизации и программирования (ОАиП)

ОТЧЕТ

по лабораторной работе №1

Тема работы: Расчет функции

Выполнил

студент: гр. 251003 Габрусь С.П.

Проверил: Фадеева Е.П.

Минск 2022

**СОДЕРЖАНИЕ**

[1 Постановка задачи 3](#_Toc81231046)

[2 Текстовый алгоритм решения задачи 4](#_Toc81231047)

[3 Структура данных 5](#_Toc81231048)

[4 Схема алгоритма решения задачи по ГОСТ 19.701-90 6](#_Toc81231049)

[5 Результаты расчетов 7](#_Toc81231050)

[Приложение А 8](#_Toc81231051)

[Приложение Б 9](#_Toc81231052)

# Постановка задачи

Разработать программу для расчета следующего матричного выражения:

Матрицы A и B – квадратные и имеют порядок равный трем – заполняются с клавиатуры.

Также необходимо вывести промежуточные и окончательные результаты расчетов.

Программа должна иметь процедуры без использования глобальных параметров. Результаты расчетов проверить в программном средстве Mathcad.

# Методика решения

Для решения поставленной задачи необходимо реализовать подпрограммы, которые выполняют четыре линейные операции над матрицами:

* сложение матриц;
* вычитание матриц;
* умножение матрицы на число;
* умножение матриц.

Рассмотрим реализацию каждой из этих подпрограмм.

1. Сложение матриц.

Чтобы сложить матрицы A и B, необходимо, чтобы они были одного размера (то есть число строк и столбцов матрицы A должно быть равно числу строк и столбцов матрицы B; в общем случае они равны m и n соответственно). Результатом сложения является матрица C такой же размерности, что и матрицы A и B. Каждый элемент матрицы C определяется следующим образом:



Для расчета суммы используется два цикла, обеспечивая проход по всем значениям матриц.

1. Вычитание матриц.

При вычитании, по аналогии со сложением, матрицы должны быть одного размера. Результатом вычитания является матрица C, каждый элемент которой определяется следующим образом:



Используемые циклы аналогичны с методом расчета суммы.

1. Умножение матрицы на число.

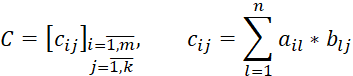
Произведением матрицы A на число N называется матрица C, каждый элемент которой – это исходное значение матрицы A, умноженное на N:



Используемые циклы аналогичны с методом расчета суммы.

1. Умножение матриц.

Для умножения матриц необходимо, чтобы матрицы были совместимыми. Матрицы A и B называются совместимыми, если число столбцов матрицы Am×n (т.е. n) совпадает с числом строк матрицы Bn×k (т.е. n). Результатом умножения матриц A и B является матрица Cm×k, каждый элемент которой определяется следующим образом:



Для расчета произведения матриц используются три цикла, два из которых определяют номер строки или столбца результирующей матрицы, а третий обеспечивает проход по значениям строк и столбцов матриц в зависимости от направления движения.

# Описание алгоритмов решения задачи

Таблица 1 – Описание алгоритмов решения задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  п.п. | Наименование  алгоритма | Назначение  алгоритма | Формальные параметры | Предполагаемый тип реализации |
| 1 | Основной алгоритм | Вызов следующих подпрограмм:  Input,  SumMatrixes,  SubMatrixes,  MultConstMatrix,  MultMatrixes,  Output |  |  |
| 2 | Input  (a, b) | Заполнение матриц данными, введенными с клавиатуры, проверка данных на корректность | а - получает от фактического параметра адрес c защитой;  b - получает от фактического параметра адрес c защитой | Процедура |
| 3 | SumMatrixes  (a, b, result) | Суммирует матрицы a и b с сохранением результата в матрице result | а - получает от фактического параметра адрес c защитой;  b - получает от фактического параметра адрес c защитой;  result - получает от фактического параметра адрес, возвращаемый параметр | Функция.  result – возвращаемый функцией параметр |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | SubMatrixes  (a, b, result) | Вычитает матрицу b из матрицы a с сохранением результата в result | а - получает от фактического параметра адрес c защитой;  b - получает от фактического параметра адрес c защитой;  result - получает от фактического параметра адрес, возвращаемый параметр | Функция.  result – возвращаемый функцией параметр |
| 5 | MultConstMatrix  (Numb, a,  result) | Умножает матрицу a на число Numb с сохранением результата в result | Numb - получает от фактического параметра адрес c защитой;  a - получает от фактического параметра адрес c защитой;  result - получает от фактического параметра адрес, возвращаемый параметр | Функция.  result – возвращаемый функцией параметр |
| 6 | MultMatrixes  (a, b, result) | Производит умножение матрицы a на матрицу b с сохранением результата в result | а - получает от фактического параметра адрес c защитой;  b - получает от фактического параметра адрес c защитой;  result - получает от фактического параметра адрес, возвращаемый параметр | Функция.  result – возвращаемый функцией параметр |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 | Output  (a) | Выводит матрицу a на экран | а - получает от фактического параметра адрес c защитой | Процедура |

# Структура данных

Таблица 2 – Структура данных основного алгоритма

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| A, B | array [1..3, 1..3] of Integer | Исходные матрицы |
| RES | array [1..3, 1..3] of Integer | Матрица, хранящая результат вычисления выражения |

Таблица 3 – Структура данных алгоритма Input (a, b)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| i, j | Byte | Счетчики циклов | Локальный |
| isCorrectInput | Boolean | Индикатор корректности вводимых данных | Локальный |
| a, b | Array [1..3, 1..3] of Integer | Матрицы для заполнения с клавиатуры | Формальный |

Таблица 4 – Структура данных алгоритма SumMatrixes (a, b, result)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| i, j | Byte | Счетчики циклов | Локальный |
| a, b | Array [1..3, 1..3] of Integer | Исходные матрицы | Формальный |
| result | Array [1..3, 1..3] of Integer | Матрица с результатами расчета | Формальный |

Таблица 5 – Структура данных алгоритма SubMatrixes (a, b, result)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| i, j | Byte | Счетчики циклов | Локальный |
| a, b | Array [1..3, 1..3] of Integer | Исходные матрицы | Формальный |
| result | Array [1..3, 1..3] of Integer | Матрица с результатами расчета | Формальный |
|  |  |  |  |

Таблица 6 – Структура данных алгоритма MultConstMatrix (numb, a, result)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| i, j | Byte | Счетчики циклов | Локальный |
| Numb | Interger | Исходное число | Формальный |
| a | Array [1..3, 1..3] of Integer | Исходная матрица | Формальный |
| result | Array [1..3, 1..3] of Integer | Матрица с результатами расчета | Формальный |

Таблица 7 – Структура данных алгоритма MultMatrixes (a, b, result)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| i, j, k | Byte | Счетчики циклов | Локальный |
| a, b | Array [1..3, 1..3] of Integer | Исходные матрицы | Формальный |
| result | Array [1..3, 1..3] of Integer | Матрица с результатами расчета | Формальный |

Таблица 8 – Структура данных алгоритма Output (a)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| i, j | Byte | Счетчики циклов | Локальный |
| a | Array [1..3, 1..3] of Integer | Матрицы, для вывода на экран | Формальный |

# Схема алгоритма решения задачи по ГОСТ 19.701-90

## Схема основного алгоритма



Рисунок 1 – Схема основного алгоритма

## Схема алгоритма Input



Рисунок 2 – Схема алгоритма Input

## Схема алгоритма SumMatrixes



Рисунок 3 – Схема алгоритма SumMatrixes

## Схема алгоритма SubMatrixes



Рисунок 4 – Схема алгоритма SubMatrixes

## Схема алгоритма MultConstMatrix



Рисунок 5 – Схема алгоритма MultConstMatrixes

## Схема алгоритма MultMatrixes



Рисунок 6 – Схема алгоритма MultMatrixes

## Схема алгоритма Output



Рисунок 7 – Схема алгоритма Output

Приложение А

(обязательное)

Исходный код программы

Program lab1FinalAnswer;

{$APPTYPE CONSOLE}

{

Using the functions (procedures) the program is need

to calculate the next expression:

2 \* (A + B) \* (2 \* B - A)

}

Uses

System.SysUtils;

//Determining the type for matrixes

Type

TMatrix = array [1..3, 1..3] of Integer;

Var

A, B, RES: TMatrix;

//A - first input matrix

//B - second input matrix

//RES - result of the expression

//i, j - iterators for the cycles

//Procedure for input the matrixes

Procedure Input(var a, b: TMatrix);

var

i, j: byte;

isCorrectInput: Boolean;

//i, j - iterators for cycles

//isCorrectInput - boolean variable to check input

//for correctness

Begin

writeln('You need to input matrixes and then the next expression will be calculated: 2 \* (A + B) \* (2 \* B - A).');

writeln('(the elemets of matrixes must be integer in the range [-2147483647, 2147483647])');

writeln('Enter the matrix A:');

//Repeating input matrix A until it's correct

repeat

//Making the input correct by default

isCorrectInput := true;

//Inputing the matrix A

for i := 1 to 3 do

for j := 1 to 3 do

begin

//Checking for exceptions

try

read(a[i][j]);

except

isCorrectInput := false;

end;

end;

writeln;

//Telling the user that input is incorrect

if not isCorrectInput then

writeln('Oops, your input is incorrect. Try again:');

until isCorrectInput;

writeln('Enter the matrix B:');

//Repeating input matrix B until it's correct

repeat

//Making the input correct by default

isCorrectInput := true;

//Inputing the matrix B

for i := 1 to 3 do

for j := 1 to 3 do

begin

//Checking for exceptions

try

read(b[i][j]);

except

isCorrectInput := false;

end;

end;

writeln;

//Telling the user that input is incorrect

if not isCorrectInput then

writeln('Oops, your input is incorrect. Try again:');

until isCorrectInput;

End;

//Procedure for summing two matrixes

Function SumMatrixes (const a, b: TMatrix): TMatrix;

var

i, j: byte;

//i, j - iterators for cycles

Begin

//Summing the matrixes

for i := 1 to 3 do

for j := 1 to 3 do

result[i][j] := a[i][j] + b[i][j];

End;

//Procedure for substract two matrixes

Function SubMatrixes (const a, b: TMatrix): TMatrix;

var

i, j: byte;

//i, j - iterators for cycles

Begin

//Substract the matrixes

for i := 1 to 3 do

for j := 1 to 3 do

result[i][j] := a[i][j] - b[i][j];

End;

//Procedure for multiplying number on matrix

Function MultConstMatrix(const Numb: integer;

const a: TMatrix): TMatrix;

var

i, j: byte;

//i, j - iterators for cycles

Begin

//Myltiplying number on matrix

for i := 1 to 3 do

for j := 1 to 3 do

result[i][j] := Numb \* a[i][j];

End;

//Procedure for multiplying matrixes

Function MultMatrixes(const a, b: TMatrix): TMatrix;

var

i, j, k: byte;

//i, j, k - iterators for cycles

Begin

//Myltiplying the matrixes

for i := 1 to 3 do

for j := 1 to 3 do

begin

result[i][j] := 0;

for k := 1 to 3 do

result[i][j] := result[i][j] + a[i][k] \* b[k][j];

end;

End;

//Procedure for output the result

Procedure Output(const a: TMatrix);

var

i, j: byte;

//i, j - iterators for cycles

Begin

for i := 1 to 3 do

begin

for j := 1 to 3 do

write(a[i][j]:11);

writeln;

end;

End;

Begin

//Inputing the matrixes

Input(A, B);

//Counting the result

RES := MultMatrixes(MultConstMatrix(2,

SumMatrixes(A, B)),

SubMatrixes(MultConstMatrix(2, B), A));

//Outputing the result

writeln('2 \* (A + B) \* (2 \* B - A) = ');

Output(RES);

readln;

readln;

End.

Приложение Б

(обязательное)

Тестовые наборы

Группа тестов № 1. Некорректный ввод

Тест 1

Исходные данные: A =

1 asd 5

12 -23 6

0 45 24

Ожидаемый результат: Некорректный ввод.

Полученный результат:

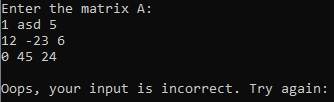


Рисунок 8 – Тестовые наборы. Некорректный ввод

Тест 2

Исходные данные: A =

1 572 5

12 -23 6

0 4.5 24

Ожидаемый результат: Некорректный ввод.

Полученный результат:

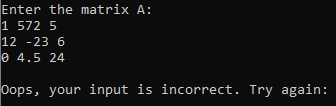


Рисунок 9 – Тестовые наборы. Некорректный ввод

Тест 3

Исходные данные: A =

1 572 5000000000

12 -23 6

0 45 24

Ожидаемый результат: Некорректный ввод.

Полученный результат:

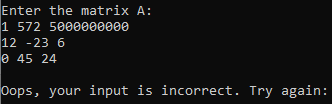


Рисунок 10 – Тестовые наборы. Некорректный ввод

Группа тестов № 2. Корректный ввод

Таблица 9 – Тестовые наборы

|  |  |  |
| --- | --- | --- |
| Тест | Исходные данные и ожидаемый  результат (Mathcad) | Полученный результат |
| 1. |  |  |
| 2. |  |  |

Продолжение таблицы 9

|  |  |  |
| --- | --- | --- |
| 3. |  |  |

Приложение В

(обязательное)

Исходный код программы с промежуточными ответами

Program lab1InterimAnswers;

{$APPTYPE CONSOLE}

{

Using the functions (procedures) the program need to

calculate the next expression:

2 \* (A + B) \* (2 \* B - A)

}

uses

System.SysUtils;

//Determining the type for matrixes

Type

TMatrix = array [1..3, 1..3] of Integer;

Var

A, B, Temp1, Temp2, RES: TMatrix;

//A - first input matrix

//B - second input matrix

//Temp1, Temp2 - matrix for interim results

//RES - matrix for the final result

//Procedure for input the matrixes

Procedure Input(var a, b: TMatrix);

var

i, j: byte;

isCorrectInput: boolean;

//i, j - iterators for the cycles

//isCorrectInput - boolean variable to check input

//for correctness

Begin

writeln('You need to input matrixes and then the next expression will be calculated: 2 \* (A + B) \* (2 \* B - A).');

writeln('(the elemets of matrixes must be integer in the range [-2147483647, 2147483647])');

writeln('Enter the matrix A:');

//Repeating input matrix A until it's correct

repeat

//Making the input correct by default

isCorrectInput := true;

//Inputing the matrix A

for i := 1 to 3 do

for j := 1 to 3 do

begin

//Checking for exceptions

try

read(a[i][j]);

except

isCorrectInput := false;

end;

end;

writeln;

//Telling the user that input is incorrect

if not isCorrectInput then

writeln('Oops, your input is incorrect. Try again:');

until isCorrectInput;

writeln('Enter the matrix B:');

//Repeating input matrix B until it's correct

repeat

//Making the input correct by default

isCorrectInput := true;

//Inputing the matrix B

for i := 1 to 3 do

for j := 1 to 3 do

begin

//Checking for exceptions

try

read(b[i][j]);

except

isCorrectInput := false;

end;

end;

writeln;

//Telling the user that input is incorrect

if not isCorrectInput then

writeln('Oops, your input is incorrect. Try again:');

until isCorrectInput;

End;

//Procedure for summing two matrixes

Function SumMatrixes (const a, b: TMatrix): TMatrix;

var

i, j: byte;

//i, j - iterators for the cycles

Begin

//Summing the matrixes

for i := 1 to 3 do

for j := 1 to 3 do

result[i][j] := a[i][j] + b[i][j];

End;

//Procedure for substract two matrixes

Function SubMatrixes (const a, b: TMatrix): TMatrix;

var

i, j: byte;

//i, j - iterators for the cycles

Begin

//Substract the matrixes

for i := 1 to 3 do

for j := 1 to 3 do

result[i][j] := a[i][j] - b[i][j];

End;

//Procedure for multiplying number on matrix

Function MultConstMatrix(const Numb: byte;

const a: TMatrix): TMatrix;

var

i, j: byte;

//i, j - iterators for the cycles

Begin

//Myltiplying number on matrix

for i := 1 to 3 do

for j := 1 to 3 do

result[i][j] := Numb \* a[i][j];

End;

//Procedure for multiplying matrixes

Function MultMatrixes(const a, b: TMatrix): TMatrix;

var

i, j, k: byte;

//i, j, k - iterators for the cycles

Begin

//Myltiplying the matrixes

for i := 1 to 3 do

for j := 1 to 3 do

begin

result[i][j] := 0;

for k := 1 to 3 do

result[i][j] := result[i][j] + a[i][k] \* b[k][j];

end;

End;

//Procedure for outputting interim results

Procedure Output(const Expression: string;

const a: TMatrix);

var

i, j: byte;

//i, j - iterators for the cycles

Begin

writeln(Expression);

for i := 1 to 3 do

begin

for j := 1 to 3 do

write(a[i][j]:10);

writeln;

end;

writeln;

End;

Begin

//Inputing the matrixes

Input(A, B);

//A + B

Temp1 := SumMatrixes(A, B);

Output('A + B =', Temp1);

//2 \* (A + B)

Temp1 := MultConstMatrix(2, Temp1);

Output('2 \* (A + B) = ', Temp1);

//2 \* B

Temp2 := MultConstMatrix(2, B);

Output('2 \* B = ', Temp2);

//2 \* B - A

Temp2 := SubMatrixes(Temp2, A);

Output('2 \* B - A = ', Temp2);

//2 \* (A + B) \* (2 \* B - A)

RES := MultMatrixes(Temp1, Temp2);

Output('2 \* (A + B) \* (2 \* B - A) =', RES);

readln;

readln;

End.