

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
Национальный исследовательский университет «МЭИ»

Типовой расчет «Процедуры, функции, модули»
Вариант 22

Задание выполнил: Сафутин С.В.
Студент группы А-04-18

Проверил: _____

Оценка: _____

Замечания:

Спецификация к типовому расчету

"Процедуры, функции, модули"

по дисциплине «Программирование»

1 семестр

(Вариант 22)

Абстракция А0

1. Постановка задачи.

Задание:

Написать программу обработки двух матриц в соответствии с условием.

Условие:

Дана матрица A из N строк и N столбцов.

Кроме матрицы A дана матрица B такого же размера.

Если каждый элемент матрицы A больше соответствующего элемента матрицы B ,

присвоить элементам C_1, C_2, \dots, C_N значения по правилу:

$$C_i = \sum_{k=1}^N (A_{ik} + B_{ik}) (*)$$

2. Уточненная постановка задачи.

Дана две квадратные матрицы A и B , состоящие из N строк и N столбцов.

Сравнить соответствующие элементы матриц A и B .

Если $A_{ij} > B_{ij}, \forall i, j: 0 < i < N + 1, 0 < j < N + 1$, то

сформировать одномерный массив C_1, C_2, \dots, C_N по правилу (*).

3. Пример с иллюстрацией.

```
In [4]: from IPython.display import Image
import numpy
import pylab
%matplotlib inline
pylab.style.use('seaborn-whitegrid')
```

```
In [24]: def write_datatxt(file_name, N, A, B):
    with open(file_name, 'w') as f:
        for M in [A, B]:
            for i in range(N):
                f.write("{}\n".format(M[i]))
```

```
In [25]: N1 = 5
A1 = numpy.random.randint(0, 100, size=(N1,N1))
B1 = numpy.random.randint(-100, 0, size=(N1,N1))
write_datatxt('AB_SC_1.txt', N1, A1, B1)
print('\n', N1, '\n\n', A1, '\n\n', B1)
```

5

```
[[38 74 64 76 99]
 [53 98 15 33 45]
 [91 49 97 84 90]
 [22 48 83 0 63]
 [99 52 33 99 48]]
```

```
[[ -76 -98 -99 -98 -82]
 [-16 -22 -65 -83 -75]
 [-25 -76 -66 -8 -2]
 [-70 -45 -74 -70 -43]
 [-37 -84 -64 -3 -94]]
```

```
In [26]: N2 = 5
A2 = numpy.random.randint(1, 3, size=(N2,N2))
B2 = numpy.random.randint(-2, 0, size=(N2,N2))
write_datatxt('AB_SC_2.txt', N2, A2, B2)
print('\n', N2, '\n\n', A2, '\n\n', B2)
```

5

```
[[2 1 1 2 1]
 [2 2 2 1 1]
 [1 1 1 1 1]
 [1 1 1 1 2]
 [1 2 1 1 2]]
```

```
[[ -1 -1 -1 -1 -1]
 [-1 -1 -1 -2 -1]
 [-2 -1 -2 -2 -1]
 [-1 -1 -2 -2 -2]
 [-1 -2 -1 -2 -2]]
```

```
In [27]: N3 = 5
A3 = numpy.random.randint(0, 2, size=(N3,N3))
B3 = numpy.random.randint(0, 2, size=(N3,N3))
write_datatxt('AB_SC_3.txt', N3, A3, B3)
print('\n', N3, '\n\n', A3, '\n\n', B3)
```

5

```
[[1 1 0 0 0]
 [0 0 0 1 1]
 [0 0 0 0 0]
 [0 1 1 0 0]
 [1 1 0 0 1]]
```

```
[[0 0 1 1 0]
 [1 1 1 0 0]
 [0 1 0 1 0]
 [1 1 0 0 0]
 [0 0 1 1 1]]
```

```
In [28]: N4 = 5
A4 = numpy.random.randint(1, 10, size=(N4,N4))
B4 = numpy.random.randint(-10, 0, size=(N4,N4))
write_datetxt('AB_SC_4.txt', N4, A4, B4)
print('\n', N4, '\n\n', A4, '\n\n', B4)

5

[[7 6 9 7 4]
 [6 9 2 1 2]
 [3 1 2 7 5]
 [3 4 8 3 1]
 [3 6 2 1 2]]

[[ -9  -8  -2  -9 -10]
 [ -9  -6  -3  -7  -8]
 [-10 -10  -1  -7  -9]
 [ -6  -7  -3  -9  -8]
 [ -2  -9  -8  -7  -5]]
```

```
In [29]: N5 = 5
A5 = numpy.random.randint(0, 10, size=(N5,N5))
B5 = numpy.random.randint(0, 9, size=(N5,N5))
write_datetxt('AB_SC_5.txt', N5, A5, B5)
print('\n', N5, '\n\n', A5, '\n\n', B5)

5

[[5 1 8 2 8]
 [4 3 6 0 8]
 [0 0 3 8 5]
 [5 3 1 2 3]
 [1 3 9 8 0]]

[[5 0 7 6 7]
 [7 3 6 2 5]
 [1 6 6 1 4]
 [2 4 8 4 4]
 [4 5 3 0 0]]
```

```
In [60]: print(20*' ' + 'Лаб.12 Типовой Расчет \n' + 90*'=')

fin = str(input('Текстовый файл с элементами исходных матриц: '))
fout = str(input('Текстовый файл с элементами исходных матриц и результатами: '))

#fin = 'AB_SC_1.txt', 'AB_SC_2.txt', 'AB_SC_3.txt', 'AB_SC_4.txt', 'AB_SC_5.txt'
#fout = 'ABC_SC_1.txt', 'ABC_SC_2.txt', 'ABC_SC_3.txt', 'ABC_SC_4.txt', 'ABC_SC_5.txt'

with open(fin, 'r') as f:
    Y = f.readlines()
    Y = [x[1:-2].split() for x in Y]
    Y = [[int(x) for x in Y[i]] for i in range(len(Y))]

    A, B = Y[:int(len(Y)/2)], Y[int(len(Y)/2):]

    print('Исходная матрица A: \n' + 90*'= ' + '\n')
    for i in range(len(A)):
        print("{}\n".format(A[i]))

    print('Исходная матрица B: \n' + 90*'= ' + '\n')
    for i in range(len(B)):
        print("{}\n".format(B[i]))

    C = [A[i][j] > B[i][j] for i in range(len(A)) for j in range(len(A))]
    if sum(C) == len(C):
        C = [sum([A[i][j] + B[i][j] for j in range(len(A))]) for i in range(len(A))]
        print('Сформированный массив C: \n' + 90*'= ' + '\n' + str(C))
    else:
        print('Не все элементы матрицы A больше соответствующих элементов матрицы B')

    with open(fout, 'w') as f:
        for i in range(len(A)):
            f.write("{}\n".format(A[i]))
        for i in range(len(B)):
            f.write("{}\n".format(B[i]))
        f.write("{}\n".format(C))
```

Лаб.12 Типовой Расчет

```
=====
Текстовый файл с элементами исходных матриц: AB_SC_1.txt
Текстовый файл с элементами исходных матриц и результатами: ABC_SC_1.txt
Исходная матрица A:
=====
```

```
[38, 74, 64, 76, 99]
[53, 98, 15, 33, 45]
[91, 49, 97, 84, 90]
[22, 48, 83, 0, 63]
[99, 52, 33, 99, 48]
```

Исходная матрица B:

```
=====
[-76, -98, -99, -98, -82]
[-16, -22, -65, -83, -75]
[-25, -76, -66, -8, -2]
[-70, -45, -74, -70, -43]
[-37, -84, -64, -3, -94]
```

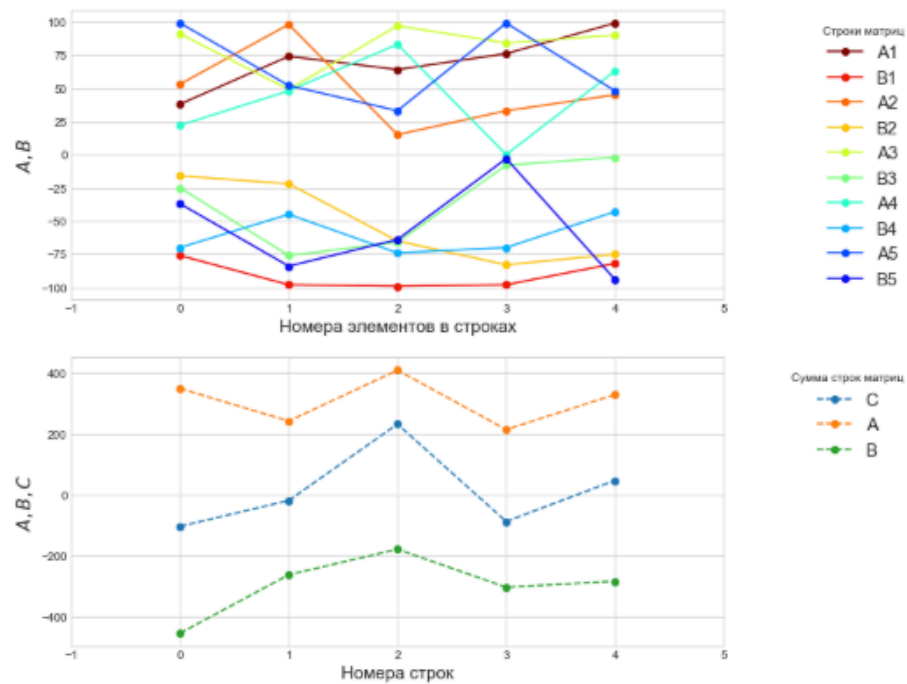
Сформированный массив C:

```
=====
[-102, -17, 234, -86, 49]
```

```
In [126]: f = pylab.figure(figsize=(10,10))
ax1 = f.add_subplot(2, 1, 1)
ax2 = f.add_subplot(2, 1, 2)
bwr = pylab.cm.get_cmap('jet_r')

for i in range(len(B)):
    ax1.plot(range(len(A[i])), A[i], '-o',
             c=bwr(2*i/10), label='A' + str(i+1))
    ax1.plot(range(len(B[i])), B[i], '-o',
             c=bwr((2*i+1)/10), label='B' + str(i+1))
ax1.legend(title='Строки матриц', fontsize=15,
          shadow=True, bbox_to_anchor=(1.3,1.0))
ax1.set_xticks(range(-1, len(C)+1), range(len(C)+2))
ax1.set_xlabel('Номера элементов в строках', fontsize=15)
ax1.set_ylabel(r'$A, B$', fontsize=15)

ax2.plot(range(len(C)), C, '-o', label='C', ls='--')
ax2.plot(range(len(C)), [sum(A[i]) for i in range(len(C))],
        '-o', label='A', ls='--')
ax2.plot(range(len(C)), [sum(B[i]) for i in range(len(C))],
        '-o', label='B', ls='--')
ax2.legend(title='Сумма строк матриц', fontsize=15,
          shadow=True, bbox_to_anchor=(1.3,1.0))
ax2.set_xticks(range(-1, len(C)+1), range(len(C)+2))
ax2.set_xlabel('Номера строк', fontsize=15)
ax2.set_ylabel(r'$A, B, C$', fontsize=15);
```



4. Таблица данных

Класс	Имя	Описание (смысл, диапазон, точность)	Тип	Структура	Формат в/в
входные данные	fin	название текстового файла с элементами массива B	символ	строка	" +XX.txt"
входные данные	fout	название текстового файла с элементами массива B	символ	строка	" +XX.txt"
входные данные	tfin	название текстового файла с элементами массива B	символ	текстовый файл	+XX.txt
входные данные	tfout	название текстового файла с элементами массива B	символ	текстовый файл	+XX.txt
входные данные	X	вводимый из текстового файла массив, содержащий исходные матрицы	цел	двумерный массив	+XX (:5)
входные данные	A, B	исходные матрицы	цел	двумерные массивы	+XX (:5)
выходные данные	C	выводимый в текстовый файл сформированный массив	цел	одномерные массивы	+XX (:5)
промежуточные данные	i	индекс текущей строки, 1≤i≤50	цел	простая переменная	---
промежуточные данные	j	индекс текущего столбца, 1≤j≤50	цел	простая переменная	---

5. Входная форма

- обр 1.1 Текстовый файл с элементами исходных матриц: $\langle fin \rangle$
- обр 1.2 Текстовый файл с элементами исходных матриц и результатами: $\langle fout \rangle$
- обр 2.1 Исходная матрица A:
- обр 2.2 $\langle A[11] \rangle \langle A[12] \rangle \dots \langle A[1n] \rangle$
 \dots
 $\langle A[n1] \rangle \langle A[n2] \rangle \dots \langle A[nn] \rangle$
- обр 3.1 Исходная матрица B:
- обр 3.2 $\langle B[11] \rangle \langle B[12] \rangle \dots \langle B[1n] \rangle$
 \dots
 $\langle B[n1] \rangle \langle B[n2] \rangle \dots \langle B[nn] \rangle$

6. Выходная форма

- обр 4 Лаб. 12 Типовой расчет
- обр 5.1 Текстовый файл с элементами исходных матриц: $\langle fin \rangle$
- обр 5.2 Текстовый файл с элементами исходных матриц и результатами: $\langle fout \rangle$
- обр 6.1 Исходная матрица A:
- обр 6.2 $\langle A[11] \rangle \langle A[12] \rangle \dots \langle A[1n] \rangle$
 \dots
 $\langle A[n1] \rangle \langle A[n2] \rangle \dots \langle A[nn] \rangle$
- обр 7.1 Исходная матрица B:
- обр 7.2 $\langle B[11] \rangle \langle B[12] \rangle \dots \langle B[1n] \rangle$
 \dots
 $\langle B[n1] \rangle \langle B[n2] \rangle \dots \langle B[nn] \rangle$
- обр 8 Некорректное число строк или столбцов
- обр 9.1 Сформированный массив C:
- обр 9.2 $\langle C[1] \rangle \langle C[2] \rangle \dots \langle C[n] \rangle$

7. Аномалии

N	Описание	Условие возникновения	Реакция
1	Не входных данных	$X[i,j] = 0 \forall i,j$	Обр 8
2	Число строк не равно удвоенному числу столбцов	$i \neq 2j$	Обр 8

8. Функциональные тесты

Исходные данные	-	-	-	Результаты	-	Тест
fin	fout	A	B	C	Сообщение	№
AB_SC_1.txt	ABC_SC_1.txt	[[38, 74, 64, 76, 99] [53, 98, 15, 33, 45] [91, 49, 97, 84, 90] [22, 48, 83, 0, 63] [99, 52, 33, 99, 48]]	[-76, -98, -99, -98, -82] [-16, -22, -65, -83, -75] [-25, -76, -66, -8, -2] [-70, -45, -74, -70, -43] [-37, -84, -64, -3, -94]	[-102, -17, 234, -86, 49]	-	1
-	-	-	-	-	-	-
AB_SC_2.txt	ABC_SC_2.txt	[2, 1, 1, 2, 1] [2, 2, 2, 1, 1] [1, 1, 1, 1, 1] [1, 1, 1, 1, 2] [1, 2, 1, 1, 2]	[-1, -1, -1, -1, -1] [-1, -1, -1, -2, -1] [-2, -1, -2, -2, -1] [-1, -1, -2, -2, -2] [-1, -2, -1, -2, -2]	[2, 2, -3, -2, -1]	-	2
-	-	-	-	-	-	-
AB_SC_3.txt	ABC_SC_3.txt	[1, 1, 0, 0, 0] [0, 0, 0, 1, 1] [0, 0, 0, 0, 0] [0, 1, 1, 0, 0] [1, 1, 0, 0, 1]	[0, 0, 1, 1, 0] [1, 1, 1, 0, 0] [0, 1, 0, 1, 0] [1, 1, 0, 0, 0] [0, 0, 1, 1, 1]	-	Не все элементы матрицы A больше соответствующих элементов матрицы B	3
-	-	-	-	-	-	-
AB_SC_4.txt	ABC_SC_4.txt	[7, 6, 9, 7, 4] [6, 9, 2, 1, 2] [3, 1, 2, 7, 5] [3, 4, 8, 3, 1] [3, 6, 2, 1, 2]	[-9, -8, -2, -9, -10] [-9, -6, -3, -7, -8] [-10, -10, -1, -7, -9] [-6, -7, -3, -9, -8] [-2, -9, -8, -7, -5]	[-5, -13, -19, -14, -17]	-	4
-	-	-	-	-	-	-
AB_SC_5.txt	ABC_SC_5.txt	[5, 1, 8, 2, 8] [4, 3, 6, 0, 8] [0, 0, 3, 8, 5] [5, 3, 1, 2, 3] [1, 3, 9, 8, 0]	[5, 0, 7, 6, 7] [7, 3, 6, 2, 5] [1, 6, 6, 1, 4] [2, 4, 8, 4, 4] [4, 5, 3, 0, 0]	-	Не все элементы матрицы A больше соответствующих элементов матрицы B	5
-	-	-	-	-	-	-

№ теста	Входные данные	-	-	-	Ожидаемый результат	-	Смысл теста
-	-	-	-	-	-	-	-
1	fin = AB_sc_1.txt	fout = ABC_sc_1.txt	A = [[38, 74, 64, 76, 99] [53, 98, 15, 33, 45] [91, 49, 97, 84, 90] [22, 48, 83, 0, 63]] [99, 52, 33, 99, 48]]	B = [-76, -98, -99, -98, -82] [-16, -22, -65, -83, -75] [-25, -76, -66, -8, -2] [-70, -45, -74, -70, -43] [-37, -84, -64, -3, -94]	C = [-102, -17, 234, -86, 49]	-	Подтвердить правильность расчетов в случае широкого спектра положительных и отрицательных чисел
-	-	-	-	-	-	-	-
2	fin = AB_sc_2.txt	fout = ABC_sc_2.txt	A = [2, 1, 1, 2, 1] [2, 2, 2, 1, 1] [1, 1, 1, 1, 1] [1, 1, 1, 2, 1] [1, 2, 1, 1, 2]	B = [-1, -1, -1, -1, -1] [-1, -1, -1, -2, -1] [-2, -1, -2, -2, -1] [-1, -1, -2, -2, -2] [-1, -2, -1, -2, -2]	C = [2, 2, -3, -2, -1]	-	Протестировать простейший случай
-	-	-	-	-	-	-	-
3	fin = AB_sc_3.txt	fout = ABC_sc_3.txt	A = [1, 1, 0, 0, 0] [0, 0, 0, 1, 1] [0, 0, 0, 0, 0] [0, 1, 1, 0, 0] [1, 1, 0, 0, 1]	B = [0, 0, 1, 1, 0] [1, 1, 0, 0] [0, 1, 0, 1, 0] [1, 1, 0, 0, 0] [0, 0, 1, 1, 1]	-	Не все элементы матрицы A больше соответствующих элементов матрицы B	Протестировать простейший случай с бинарными значениями элементов
-	-	-	-	-	-	-	-
4	fin = AB_sc_4.txt	fout = ABC_sc_4.txt	A = [7, 6, 9, 7, 4] [6, 9, 2, 1, 2] [3, 1, 2, 7, 5] [3, 4, 8, 3, 1] [3, 6, 2, 1, 2]	B = [-9, -8, -2, -9, -10] [-9, -6, -3, -7, -8] [-10, -10, -1, -7, -9] [-6, -7, -3, -9, -8] [-2, -9, -8, -7, -5]	C = [-5, -13, -19, -14, -17]	-	Протестировать массив в узком диапазоне положительных и отрицательных чисел
-	-	-	-	-	-	-	-
5	fin = AB_sc_5.txt	fout = ABC_sc_5.txt	A = [5, 1, 8, 2, 8] [4, 3, 6, 0, 8] [0, 0, 3, 8, 5] [5, 3, 1, 2, 3] [1, 3, 9, 8, 0]	B = [5, 0, 7, 6, 7] [7, 3, 6, 2, 5] [1, 6, 6, 1, 4] [2, 4, 8, 4, 4] [4, 5, 3, 0, 0]	-	Не все элементы матрицы A больше соответствующих элементов матрицы B	Подтвердить правильность расчетов в узком диапазоне положительных чисел

-	Результаты	№ теста
-	-	-
C	Максимальная вычислительная нагрузка = 25 (сравнения) + 5 (суммирование)	1
-	Соответствует ожидаемым числовым значениям	-
C	Максимальная вычислительная нагрузка = 25 (сравнения) + 5 (суммирование)	2
-	Соответствует ожидаемым числовым значениям	-
C	-	3
-	Задача не решалась, массивы не соответствуют условиям	-
C	Максимальная вычислительная нагрузка = 25 (сравнения) + 5 (суммирование)	4
-	Соответствует ожидаемым числовым значениям	-
C	-	5
-	Задача не решалась, массивы не соответствуют условиям	-
-	-	-
Сообщение	-	1
Сообщение	-	2
Сообщение	Не все элементы матрицы A больше соответствующих элементов матрицы B	3
Сообщение	-	4
Сообщение	Не все элементы матрицы A больше соответствующих элементов матрицы B	5

9. Метод

Отделим ввод-вывод от обработки данных и разделим задачу на три подзадачи:

1. Подзадача А 0.1.

Запросить названия текстовых файлов ввода-вывода, считать из текстового файла ввода исходные данные (обр.1-3), затем вывести их на экран (обр.4-7) для визуального подтверждения.

Вывести сообщение о наличии некорректных данных, если задача не решена (обр.8).

2. Подзадача А 0.2. Решение поставленной задачи:

- 2.1. На основе введенных исходных данных (A, B) определить, каждый ли элемент матрицы A больше соответствующего элемента матрицы B .
- 2.2. Если условие 2.1 выполнено, то сформировать новый массив C по правилу (*).

3. Подзадача А 0.3. Ввести полученные результаты (C) на экран (обр.9.1, 9.2) и записать в файл вывода исходные матрицы и полученный результат.

Ввод-вывод тривиален и представляет собой чисто техническую задачу.

Для осуществления процесса ввода-вывода и обработки матриц потребуются промежуточные переменные:

- X - вводимый из текстового файла двумерный массив, содержащий исходны матрицы,
- n - размерность исходных матриц,
- i - индекс строки текущего элемента матрицы,
- j - индекс столбца текущего элемента матрицы.

Их следует добавить в таблицу данных.

Алгоритм необходим только для обработка входных данных и расчета результата.

10. Алгоритм

[Sample diagrams](#)

```
In [8]: %%file block_diagram

blockdiag (
    orientation = portrait;
    class start_end [shape = ellipse, fontsize = 15, color = lightskyblue];
    class input_out [shape = flowchart.input, fontsize = 15, width = 600, color = lightgray];
    class main_box [shape = box, width = 450, height = 150, fontsize = 18];

    Начало ->
    "Вывод заголовка по обр.4 на экран" -> "Ввод имен файлов по обр.1.1,1.2" ->
    "Вывод имен файлов по обр.5.1,5.2 на экран" ->
    "Процедура 1. \nСохранение исходных данных \nв виде матрицы X" ->
    "Процедура 2. \nВывод на экран и в файл \nматриц A,B" ->
    "Процедура 3. \nСравнение \nэлементов матриц A,B \nФормирование \nmассива C" ->
    "Процедура 4. \nВывод на экран и в файл \nmассива C" ->
    Конец;

    Начало, Конец [class = "start_end"];
    "Вывод заголовка по обр.4 на экран", "Ввод имен файлов по обр.1.1,1.2" [class = "input_out"];
    "Вывод имен файлов по обр.5.1,5.2 на экран" [class = "input_out"];
    "Процедура 1. \nСохранение исходных данных \nв виде матрицы X" [class = "main_box"];
    "Процедура 2. \nВывод на экран и в файл \nматриц A,B" [class = "main_box"];
    "Процедура 3. \nСравнение \nэлементов матриц A,B \nФормирование \nmассива C" [class = "main_box"];
    "Процедура 4. \nВывод на экран и в файл \nmассива C" [class = "main_box"];

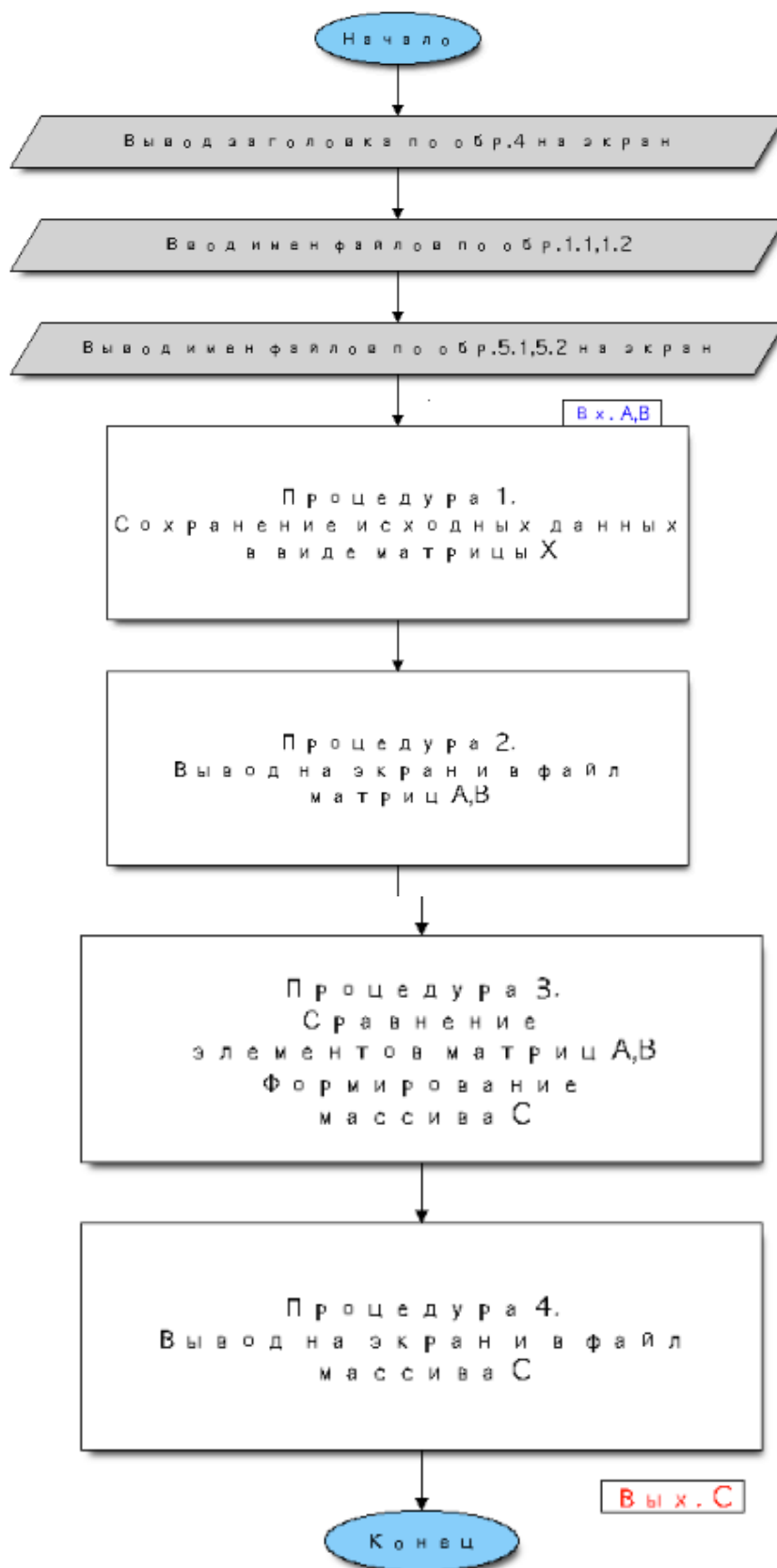
    "Вывод имен файлов по обр.5.1,5.2 на экран" ->
    "Процедура 1. \nСохранение исходных данных \nв виде матрицы X" [label = 'Bx. A,B', fontsize = 18, textcolor = blue];
    "Процедура 4. \nВывод на экран и в файл \nmассива C" ->
    Конец [label = 'Вых. C', fontsize = 18, textcolor = red];
}
```

Overwriting block_diagram

```
In [9]: !blockdiag block_diagram
```

```
In [10]: Image("block_diagram.png")
```

```
Out[10]:
```




11. Программа на *Delphi*.

Диалоговый вариант (ввод названий файлов и данных из текстового файла, вывод на экран и в текстовый файл)

Синтаксис:

[Pascal Tutorial](#)

```
In [ ]: program Lab12; // сохранить как lab12.dpr (DPR  Delphi PProject)
{$APPTYPE CONSOLE}
Uses Windows; //

type
  M1 = array [1..20,1..10] of integer;
  M2 = array [1..10,1..10] of integer;
  A1 = array [1..10] of integer;

var
  fin, fout: string;
  tfin, tfout: TextFile;
  i, j, n, t: integer;
  X: M1;
  A, B: M2;
  C: A1;

procedure read_in(var X: M1; var i, j, n: integer);
begin
  i := 1;
  while not eof(tfin) do
  begin
    j := 1;
    while not eof(tfin) do
    begin
      read(tfin, X[i,j]);
      j := j + 1;
    end;
    i := i + 1;
    readln(tfin);
  end;
  n := round((i - 1) / 2);
end;

procedure write_in(n: integer; X: M1; var A, B: M2; var i, j: integer);
begin
  writeln('Исходная матрица A');
  i := 1;
  while i < 2 * n + 1 do
  begin
    if i = n + 1 then
      writeln('Исходная матрица B');
    if i < n + 1 then
    begin
      j := 1;
      while j < n + 1 do
      begin
        A[i,j] := X[i,j];
        write(A[i,j]:5);
        write(tfout, A[i,j]:5);
        j := j + 1;
      end;
    end;
    else
    begin
      j := 1;
      while j < n + 1 do
      begin
        B[i-n,j] := X[i,j];
        write(B[i-n,j]:5);
        write(tfout, B[i-n,j]:5);
        j := j + 1;
      end;
    end;
    writeln; writeln(tfout);
    i := i + 1;
  end;
end;

procedure write_out(n: integer; C: A1; var i: integer);
begin
  writeln('Сформированный массив C');
  i := 1;
  while i < n + 1 do
  begin
    write(C[i]:5);
    write(tfout, C[i]:5);
    i := i + 1;
  end;
  writeln; writeln(tfout);
end;

begin
  setConsoleCP(1251); //
  setConsoleOutputCP(1251); //
```

```

{A0.1 – ввод-вывод входных данных}
writeln('Лаб.12 Типовой Расчет' :40); {вывод заголовка с переходом на следующую строку по обр.4}

{ввод-вывод строковой переменной – имени файла с входными данными по обр.1.1, 5.1}
writeln('Текстовый файл с элементами исходных матриц: ');
readln(fin); writeln(fin);

{ввод-вывод строковой переменной – имени файла с выходными данными по обр.1.2, 5.2}
writeln('Текстовый файл с элементами исходных матриц и результатами: ');
readln(fout); writeln(fout);

AssignFile(tfin, fin); reset(tfin); {открыть файл для чтения}
AssignFile(tfout, fout); rewrite(tfout); {открыть файл для записи}

read_in(X, i, j, n);
CloseFile(tfin); {закрывать файл для чтения}

if (i - 1 <> 2 * (j - 1)) then
begin
  writeln('Некорректное число строк или столбцов');
  exit;
end;
write_in(n, X, A, B, i, j);

{A0.2 – формирование нового массива}
{заглушка для теста 1}
C: A1 = (-102, -17, 234, -86, 49);
{заглушка для теста 2}
// C: A1 = (2, 2, -3, -2, -1);
{заглушка для теста 3}
// writeln('Не все элементы матрицы A больше соответствующих элементов матрицы B');
{заглушка для теста 4}
// C: A1 = (-5, -13, -19, -14, -17);
{заглушка для теста 5}
// writeln('Не все элементы матрицы A больше соответствующих элементов матрицы B');

for i := 1 to 40 do
  write('='); {отделим визуально чертой и строкой введенные и выводимые значения}
writeln;

{A0.3 – вывод выходных данных}
write_out(n, C, i);

CloseFile(tfout); {закрывать файл для записи}

end.

```

Раскрытие абстракции A0.2

Поскольку условие всей задачи совпадает (за исключением необходимости ввода-вывода) с условием выделенной подзадачи, вместо полной спецификации с пунктами 1-11 выполнено просто дополнение к пунктам 9-11.

9. Метод

Пусть

- i – номер текущей строки,
- j – номер текущего столбца,
- n – количество строк и столбцов исходных матриц,
- $X[i, j]$ – элемент матрицы $X[2n \times n]$ для чтения исходных данных,
- $A[i, j], B[i, j]$ – соответствующие элементы исходных матриц $A[n \times n], B[n \times n]$,
- $C[i]$ – элемент сформированного массива $C[n \times 1]$.

Процедура 1. (Абстракции A0.1)

Исходные данные сохраняются в виде матрицы $X[2n \times n]$ с помощью циклов по строкам и столбцам.

Или выдается сообщение о некорректных данных.

Процедура 2. (Абстракции A0.1)

Формируются матрицы $A[n \times n], B[n \times n]$ и выводятся на экран и в файл с выходными данными.

Процедура 3. (Абстракции A0.2)

Сравниваются соответствующие элементы матриц $A[n \times n], B[n \times n]$.

Формируется новый массив $C[n \times 1]$ по правилу (*).

Процедура 4. (Абстракции A0.3)

Сформированный массив $C[n \times 1]$ выводится на экран и записывается в файл с выходными данными.

10. Алгоритм

[Sample diagrams](#)

Процедура 3

```
In [16]: %%file block_diagram2

blockdiag {
    orientation = portrait;

    class yes {thick, label = "Да", color = blue, textcolor = blue};
    class no {thick, label = "Нет", color = red, textcolor = red};
    class start_end {shape = ellipse, fontsize = 18, color = lightsteelblue};
    class input_out {shape = flowchart.input, fontsize = 18,
        width = 400, color = lightgray};
    class loop_in {shape = flowchart.loopin, fontsize = 18,
        width = 300, color = lightsteelblue};
    class loop_out {shape = flowchart.loopout, fontsize = 18,
        width = 300, color = lightsteelblue};
    class condition {shape = flowchart.condition, fontsize = 18,
        width = 400, height = 100, color = lightsteelblue};
    class command_box {shape = roundedbox, fontsize = 18, width = 300};

    Начало -> "i := 1" -> "while i < n + 1" ->
    "j := 1; C[i] := 0" -> "while j < n + 1" ->
    "A[i,j] > B[i,j]" -> "C[i] := C[i] + A[i,j] + B[i,j]" ->
    "j := j + 1" -> "i := i + 1" ->
    "j = n" -> "i = n" -> Конеч;

    "A[i,j] > B[i,j]" -> Сообщение -> Конеч;

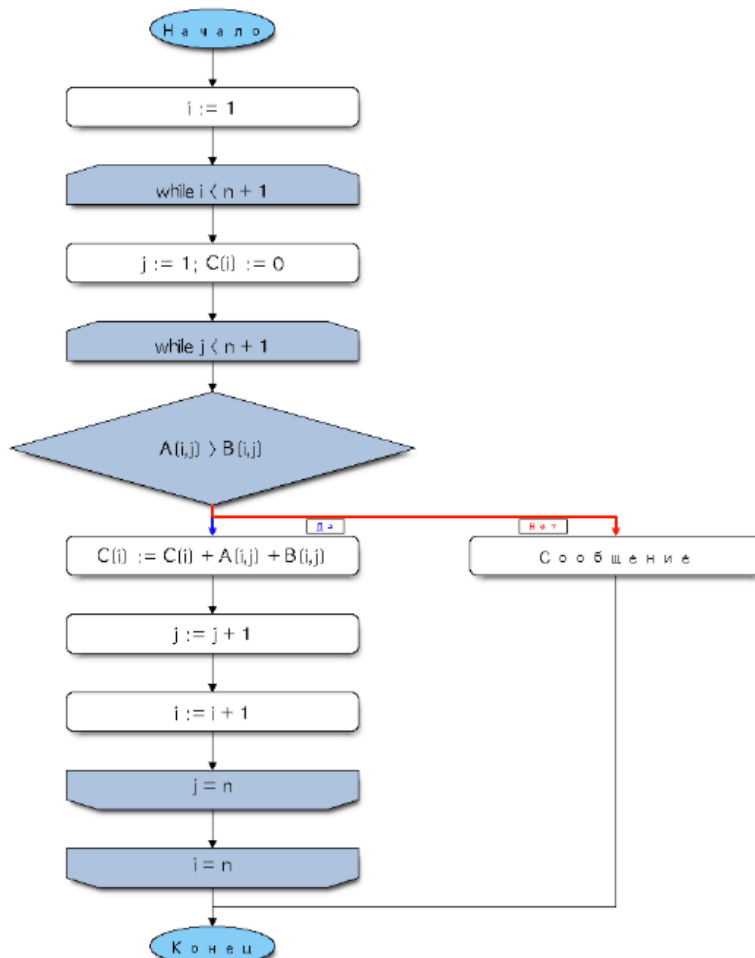
    Начало, Конеч [class = "start_end"];
    "while i < n + 1", "while j < n + 1" [class = "loop_in"];
    "i = n", "j = n" [class = "loop_out"];
    "A[i,j] > B[i,j]" [class = "condition"];
    "i := 1", "j := 1; C[i] := 0" [class = "command_box"];
    "j := j + 1", "i := i + 1" [class = "command_box"];
    "C[i] := C[i] + A[i,j] + B[i,j]", Сообщение [class = "command_box"];

    "A[i,j] > B[i,j]" -> "C[i] := C[i] + A[i,j] + B[i,j]" [class = 'yes'];
    "A[i,j] > B[i,j]" -> Сообщение [class = 'no'];
}
```

Overwriting block_diagram2

```
In [17]: !blockdiag block_diagram2
```

```
In [18]: Image("block_diagram2.png")
```



11. Программа на *Delphi*.


Программный код раскрытия абстракции.

```
In [ ]: procedure new_array(n: integer; A, B: M2; var C: A1; var i, j: integer);
begin
  i := 1;
  while i < n + 1 do
    begin
      j := 1; C[i] := 0;
      while j < n + 1 do
        begin
          if A[i,j] > B[i,j] then
            C[i] := C[i] + A[i,j] + B[i,j]
          else
            begin
              writeln('Не все элементы матрицы A больше соответствующих элементов матрицы B');
              exit;
            end;
          j := j + 1;
        end;
      i := i + 1;
    end;
  end;
end;
```

Диалоговый вариант (ввод названий файлов и данных из текстового файла, вывод на экран и в текстовый файл)

Синтаксис:

[Pascal Tutorial](#)

```
In [ ]: program Lab12; // сохранить как lab12.dpr (DPR  Delphi Project)
{$APPTYPE CONSOLE}
Uses Windows; //
type
  M1 = array [1..20,1..10] of integer;
  M2 = array [1..10,1..10] of integer;
  A1 = array [1..10] of integer;

var
  fin, fout: string;
  tf, tfout: TextFile;
  i, j, n, t: integer;
  X: M1;
  A, B: M2;
  C: A1;

procedure read_in(var X: M1; var i, j, n: integer);
begin
  i := 1;
  while not eof(tf) do
    begin
      j := 1;
      while not eof(tf) do
        begin
          read(tf, X[i,j]);
          j := j + 1;
        end;
      i := i + 1;
      readln(tf);
    end;
  n := round((i - 1) / 2);
end;

procedure write_in(n: integer; X: M1; var A, B: M2; var i, j: integer);
begin
  writeln('Исходная матрица A');
  i := 1;
  while i < 2 * n + 1 do
    begin
      if i = n + 1 then
        writeln('Исходная матрица B');
      if i < n + 1 then
        begin
          j := 1;
          while j < n + 1 do
            begin
              A[i,j] := X[i,j];
              write(A[i,j]:5);
              write(tfout, A[i,j]:5);
              j := j + 1;
            end;
          end;
        else
          begin
            j := 1;
            while j < n + 1 do
              begin
                B[i-n,j] := X[i,j];
                write(B[i-n,j]:5);
                write(tfout, B[i-n,j]:5);
                j := j + 1;
              end;
            end;
          end;
      writeln; writeln(tfout);
      i := i + 1;
    end;
  end;
```

```

procedure new_array(n: integer; A, B: M2; var C: A1; var i, j: integer);
begin
  i := 1;
  while i < n + 1 do
  begin
    j := 1; C[i] := 0;
    while j < n + 1 do
    begin
      if A[i,j] > B[i,j] then
        C[i] := C[i] + A[i,j] + B[i,j]
      else
        begin
          writeln('Не все элементы матрицы A больше соответствующих элементов матрицы B');
          exit;
        end;
      j := j + 1;
    end;
    i := i + 1;
  end;
end;

procedure write_out(n: integer; C: A1; var i: integer);
begin
  writeln('Сформированный массив C');
  i := 1;
  while i < n + 1 do
  begin
    write(C[i]:5);
    write(tfout, C[i]:5);
    i := i + 1;
  end;
  writeln; writeln(tfout);
end;

begin
  setConsoleCP(1251); //
  setConsoleOutputCP(1251); //

  {A0.1 - ввод-вывод входных данных}
  writeln('Лаб.12 Типовой Расчет' :40); {вывод заголовка с переходом на следующую строку по обр.4}

  {ввод-вывод строковой переменной - имени файла с входными данными по обр.1.1, 5.1}
  writeln('Текстовый файл с элементами исходных матриц: ');
  readln(fin); writeln(fin);

  {ввод-вывод строковой переменной - имени файла с выходными данными по обр.1.2, 5.2}
  writeln('Текстовый файл с элементами исходных матриц и результатами: ');
  readln(fout); writeln(fout);

  AssignFile(tfin, fin); reset(tfin); {открыть файл для чтения}
  AssignFile(tfout, fout); rewrite(tfout); {открыть файл для записи}

  read_in(X, i, j, n);
  CloseFile(tfin); {закрывать файл для чтения}

  if (i - 1 <> 2 * (j - 1)) then
  begin
    writeln('Некорректное число строк или столбцов');
    exit;
  end;

  write_in(n, X, A, B, i, j);

  {A0.2 - формирование нового массива}
  new_array(n, A, B, C, i, j);
  for i:=1 to 40 do
    write('='); {отделим визуально чертой и строкой введенные и выводимые значения}
  writeln;

  {A0.3 - вывод выходных данных}
  write_out(n, C, i);

  CloseFile(tfout); {закрывать файл для записи}

end.

```

Вариант программы с именами файлов, заданными в качестве параметров.

```

In [ ]: program Lab12; // сохранить как lab12.dpr (DPR H Delphi PReject)
{$APPTYPE CONSOLE}
Uses Windows; //
type
  M1 = array [1..20,1..10] of integer;
  M2 = array [1..10,1..10] of integer;
  A1 = array [1..10] of integer;

var
  tfin, tfout: TextFile;
  i, j, n: integer;
  X: M1;
  A, B: M2;
  C: A1;

```

```

procedure read_in(var X: M1; var i, j, n: integer);
begin
    i := 1;
    while not eof(tfin) do
        begin
            j := 1;
            while not eof(tfin) do
                begin
                    read(tfin, X[i,j]);
                    j := j + 1;
                end;
            i := i + 1;
            readln(tfin);
        end;
        n := round((i - 1) / 2);
    end;

procedure write_in(n: integer; X: M1; var A, B: M2; var i, j: integer);
begin
    writeln('Исходная матрица A');
    i := 1;
    while i < 2 * n + 1 do
        begin
            if i = n + 1 then
                writeln('Исходная матрица B');
            if i < n + 1 then
                begin
                    j := 1;
                    while j < n + 1 do
                        begin
                            A[i,j] := X[i,j];
                            write(A[i,j]:5);
                            write(tfout, A[i,j]:5);
                            j := j + 1;
                        end;
                    end;
                else
                    begin
                        j := 1;
                        while j < n + 1 do
                            begin
                                B[i-n,j] := X[i,j];
                                write(B[i-n,j]:5);
                                write(tfout, B[i-n,j]:5);
                                j := j + 1;
                            end;
                        end;
                    end;
                writeln; writeln(tfout);
                i := i + 1;
            end;
        end;

procedure new_array(n: integer; A, B: M2; var C: A1; var i, j: integer);
begin
    i := 1;
    while i < n + 1 do
        begin
            j := 1; C[i] := 0;
            while j < n + 1 do
                begin
                    if A[i,j] > B[i,j] then
                        C[i] := C[i] + A[i,j] + B[i,j]
                    else
                        begin
                            writeln('Не все элементы матрицы A больше соответствующих элементов матрицы B');
                            exit;
                        end;
                    j := j + 1;
                end;
            i := i + 1;
        end;
    end;

procedure write_out(n: integer; C: A1; var i: integer);
begin
    writeln('Сформированный массив C');
    i := 1;
    while i < n + 1 do
        begin
            write(C[i]:5);
            write(tfout, C[i]:5);
            i := i + 1;
        end;
    writeln; writeln(tfout);
end;

begin
    setConsoleCP(1251); //
    setConsoleOutputCP(1251); //

    {A0.1 - ввод-вывод входных данных}
    writeln('Лаб.12 Типовой Расчет' :40); {вывод заголовка с переходом на следующую строку по обр.4}

    if (ParamCount<2) then
        begin
            writeln('CritError: Не заданы параметры программы');
            readln; exit;
        end;

    AssignFile(tfin, ParamStr(1)); reset(tfin); {открыть файл для чтения}
    AssignFile(tfout, ParamStr(2)); rewrite(tfout); {открыть файл для записи}

    read_in(X, i, j, n);
    CloseFile(tfin); {закрывать файл для чтения}

```

```

    if (i - 1 <> 2 * (j - 1)) then
    begin
        writeln('Некорректное число строк или столбцов');
        exit;
    end;

    write_in(n, X, A, B, i, j);

    {A0.2 - формирование нового массива}
    new_array(n, A, B, C, i, j);
    for i:=1 to 40 do
        write('='); {отделим визуально чертой и строкой введенные и выводимые значения}
    writeln;

    {A0.3 - вывод выходных данных}
    write_out(n, C, i);

    CloseFile(tfout); {закрывать файл для записи}

end.

```

Вариант программы для онлайн-компилятора.

[Compile and Execute Pascal Online](#)

```

In [ ]: // main.pas
program Lab12;

type
    M1 = array [1..20,1..10] of integer;
    M2 = array [1..10,1..10] of integer;
    A1 = array [1..10] of integer;

var
    i, j, n: integer;
    X: M1;
    A, B: M2;
    C: A1;

procedure read_in(var X: M1; var i, j, n: integer);

begin
    i := 1;
    while not eoln do
    begin
        j := 1;
        while not eoln do
        begin
            read(X[i,j]);
            j := j + 1;
        end;
        i := i + 1;
        readln;
    end;
    n := round((i - 1) / 2);
end;

procedure write_in(n: integer; X: M1; var A, B: M2; var i, j: integer);

begin
    writeln('Исходная матрица A');
    i := 1;
    while i < 2 * n + 1 do
    begin
        if i = n + 1 then
            writeln('Исходная матрица B');
        if i < n + 1 then
        begin
            j := 1;
            while j < n + 1 do
            begin
                A[i,j] := X[i,j];
                write(A[i,j]:5);
                j := j + 1;
            end;
        end
        else
        begin
            j := 1;
            while j < n + 1 do
            begin
                B[i-n,j] := X[i,j];
                write(B[i-n,j]:5);
                j := j + 1;
            end;
        end;
        writeln;
        i := i + 1;
    end;
end;
end;

```

```

procedure new_array(n: integer; A, B: M2; var C: A1; var i, j: integer);
begin
    i := 1;
    while i < n + 1 do
        begin
            j := 1; C[i] := 0;
            while j < n + 1 do
                begin
                    if A[i,j] > B[i,j] then
                        C[i] := C[i] + A[i,j] + B[i,j]
                    else
                        begin
                            writeln('Не все элементы матрицы A больше соответствующих элементов матрицы B');
                            exit;
                        end;
                    j := j + 1;
                end;
            i := i + 1;
        end;
    end;

procedure write_out(n: integer; C: A1; var i: integer);
begin
    writeln('Сформированный массив C');
    i := 1;
    while i < n + 1 do
        begin
            write(C[i]:5);
            i := i + 1;
        end;
    end;

begin
    {A0.1 - ввод-вывод входных данных}
    writeln('Лаб.12 Типовой Расчет' :40); {вывод заголовка с переходом на следующую строку по обр.4}

    read_in(X, i, j, n);

    if (i - 1 <> 2 * (j - 1)) then
        begin
            writeln('Некорректное число строк или столбцов');
            exit;
        end;

    write_in(n, X, A, B, i, j);

    {A0.2 - формирование нового массива}
    new_array(n, A, B, C, i, j);

    for i:=1 to 40 do
        write('='); {отделим визуально чертой и строкой введенные и выводимые значения}
        writeln;

    {A0.3 - вывод выходных данных}
    write_out(n, C, i);
end.

```

```

In [ ]: // STDIN
38 74 64 76 99
53 98 15 33 45
91 49 97 84 90
22 48 83 0 63
99 52 33 99 48
-76 -98 -99 -98 -82
-16 -22 -65 -83 -75
-25 -76 -66 -8 -2
-70 -45 -74 -70 -43
-37 -84 -64 -3 -94

```

12. Структурные тесты.

В разделе функциональные тесты не были учтены аномальные ситуации:

- входных данных нет совсем;
- на ввод не поданы две квадратные матрицы одинаковой размерности.

Фрагмент программы для учета имеющихся аномалий:

```

In [ ]: if (i - 1 <> 2 * (j - 1)) then
begin
    writeln('Некорректное число строк или столбцов');
    exit;
end;

write_in(n, X, A, B, i, j);

```

Добавим 2 функциональных теста.

№ теста	Входные данные	-	-	-	Ожидаемый результат	-	Смысл теста
-	-	-	-	-	-	-	-
1	fin = AB_SC_6.txt	fout = ABC_SC_6.txt	A = []	B = []	C = -	Сообщение = Некорректное число строк или столбцов	Аномальная ситуация - отсутствие входных данных
-	-	-	-	-	-	-	-
2	fin = AB_SC_7.txt	fout = ABC_SC_7.txt	A = [2, 1, 1, 2, 1] [2, 2, 2, 1, 1] [1, 1, 1, 1, 1] [1, 1, 1, 1, 2] [1, 2, 1, 1, 2]	B = []	C = -	Сообщение = Некорректное число строк или столбцов	Аномальная ситуация - дана только одна квадратная матрица

Замечание.

Задача решена для ввода данных без разделения строк на две матрицы.

В случае наличия пустой строки между двумерными массивами в файле с входными данными решение упрощается.

Первые две процедуры можно объединить в одну и применить для каждой матрицы отдельно.

Вариант программы с именами файлов, заданными в качестве параметров.

```
In [ ]: program Lab12; // сохранить как lab12.dpr {DPR  Delphi Project}
{$APPTYPE CONSOLE}
Uses Windows; //
type
  M2 = array [1..10,1..10] of integer;
  A1 = array [1..10] of integer;

var
  tfin, tfout: TextFile;
  i, j, n: integer;
  A, B: M2;
  C: A1;

procedure read_in(var A: M2; var i, j, n: integer);
begin
  i := 1;
  while not eof(tfin) do
    begin
      j := 1;
      while not eof(tfin) do
        begin
          read(tfin, A[i,j]);
          write(A[i,j]:5);
          write(tfout, A[i,j]:5);
          j := j + 1;
        end;
      i := i + 1;
      readln(tfin); writeln; writeln(tfout);
    end;
    n := i - 1;
  end;

procedure new_array(n: integer; A, B: M2; var C: A1; var i, j: integer);
begin
  i := 1;
  while i < n + 1 do
    begin
      j := 1; C[i] := 0;
      while j < n + 1 do
        begin
          if A[i,j] > B[i,j] then
            C[i] := C[i] + A[i,j] + B[i,j]
          else
            begin
              writeln('Не все элементы матрицы A больше соответствующих элементов матрицы B');
              exit;
            end;
          j := j + 1;
        end;
      i := i + 1;
    end;
  end;
end;
```

```

procedure write_out(n: integer; C: A1; var i: integer);
begin
  writeln('Сформированный массив C');
  i := 1;
  while i < n + 1 do
  begin
    write(C[i]:5);
    write(tfout, C[i]:5);
    i := i + 1;
  end;
  writeln; writeln(tfout);
end;

begin
  setConsoleCP(1251); //
  setConsoleOutputCP(1251); //

  {A0.1 - ввод-вывод входных данных}
  writeln('Лаб.12 Типовой Расчет' :40); {вывод заголовка с переходом на следующую строку по обр.4}

  if (ParamCount<2) then
  begin
    writeln('CritError: Не заданы параметры программы');
    readln; exit;
  end;

  AssignFile(tfin, ParamStr(1)); reset(tfin); {открыть файл для чтения}
  AssignFile(tfout, ParamStr(2)); rewrite(tfout); {открыть файл для записи}

  writeln('Исходная матрица A: ');
  read_in(A, i, j, n);
  readln; writeln; writeln(tfout);
  writeln('Исходная матрица B: ');
  read_in(B, i, j, n);

  CloseFile(tfin); {закрыть файл для чтения}

  if (i - 1 <> j - 1) then
  begin
    writeln('Некорректное число строк или столбцов');
    exit;
  end;

  {A0.2 - формирование нового массива}
  new_array(n, A, B, C, i, j);

  for i:=1 to 40 do
    write('-'); {отделим визуально чертой и строкой введенные и выводимые значения}
  writeln; writeln(tfout);

  {A0.3 - вывод выходных данных}
  write_out(n, C, i);

  CloseFile(tfout); {закрыть файл для записи}

end.

```

Вариант программы для онлайн-компилятора.

[Compile and Execute Pascal Online](#)

```

In [ ]: // main.pas
program Lab12;

type
  M2 = array [1..10,1..10] of integer;
  A1 = array [1..10] of integer;

var
  i, j, n: integer;
  A, B: M2;
  C: A1;

procedure read_in(var A: M2; var i, j, n: integer);
begin
  i := 1;
  while not eoln do
  begin
    j := 1;
    while not eoln do
    begin
      read(A[i,j]);
      write(A[i,j]:5);
      j := j + 1;
    end;
    i := i + 1;
    readln; writeln;
  end;
  n := i - 1;
end;

```

```

procedure new_array(n: integer; A, B: M2; var C: A1; var i, j: integer);
begin
    i := 1;
    while i < n + 1 do
        begin
            j := 1; C[i] := 0;
            while j < n + 1 do
                begin
                    if A[i,j] > B[i,j] then
                        C[i] := C[i] + A[i,j] + B[i,j]
                    else
                        begin
                            writeln('Не все элементы матрицы A больше соответствующих элементов матрицы B');
                            exit;
                        end;
                    j := j + 1;
                end;
            i := i + 1;
        end;
    end;

procedure write_out(n: integer; C: A1; var i: integer);
begin
    writeln('Сформированный массив C');
    i := 1;
    while i < n + 1 do
        begin
            write(C[i]:5);
            i := i + 1;
        end;
    end;

begin
    {A0.1 - ввод-вывод входных данных}
    writeln('Лаб.12 Типовой Расчет' :40); {вывод заголовка с переходом на следующую строку по обр.4}

    writeln('Исходная матрица A: ');
    read_in(A, i, j, n);
    readln; writeln;
    writeln('Исходная матрица B: ');
    read_in(B, i, j, n);

    if (i - 1 <> j - 1) then
        begin
            writeln('Некорректное число строк или столбцов');
            exit;
        end;

    {A0.2 - формирование нового массива}
    new_array(n, A, B, C, i, j);

    for i:=1 to 40 do
        write('='); {отделим визуально чертой и строкой введенные и выводимые значения}
        writeln;

    {A0.3 - вывод выходных данных}
    write_out(n, C, i);
end.

```

```

In [ ]: // STDIN
38 74 64 76 99
53 98 15 33 45
91 49 97 84 90
22 48 83 0 63
99 52 33 99 48

-76 -98 -99 -98 -82
-16 -22 -65 -83 -75
-25 -76 -66 -8 -2
-70 -45 -74 -70 -43
-37 -84 -64 -3 -94

```