Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

**Национальный исследовательский университет «МЭИ»**

**Типовой расчет «Процедуры, функции, модули»**

**Вариант 20**

Задание выполнил: Самошкин Д.Ю.

Студент группы A-12-15

Проверил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Замечания:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва 2015 НИУ «МЭИ»

**Абстракция A0.**

**1.Условие**

Дана матрица B из N строк и N столбцов. Если сумма Q положительных элементов матрицы B, превышает абсолютную величину суммы отрицательных из элементов С1, С2, ..., СM, увеличить на Q значение каждого из элементов B и C  
“из N строк и N столбцов” => квадратная

**2.Уточненная постановка задачи.**Дана целочисленная квадратная матрица **B** из **N** строк и **N** столбцов. И одномерный целочисленный массив **С** из **М** элементов. Найти сумму **Q** положительных элементов матрицы **B**. Найти модуль суммы **SummC** отрицательных элементов массива **С**. Если **Q** >| **SummC** |, то увеличить значение каждого из элементов **В** и **С**, вывести измененные матрицу и одномерный массив в файл (ситуация 1). Иначе вывести сообщение, о том, что изменений не проводилось(ситуация 2).  
  
В дальнейшем будем подразумевать под словом массив одномерный целочисленный массив С, а под словом матрица целочисленную квадратную матрицу В.

**3.Примеры**1) Пример 1.(Изменения производятся)   
Исходная матрица из 3 строк и 3 столбцов ( N = 3):

10 11 -11

**B** = 4 10 -19

40 0 1

Исходный массив из 4 элементов ( M= 4):

**C** = -20 -50 19 0

**Q** = 10 + 11 + 4 +10 + 40 + 1 = 76  
 => **Q** >| **SummC** | => Изменяем матрицу и массив  
 **SummC** = | -20 -50| = 70

Изменённая матрица:

86 87 65

80 86 57

116 76 77

Изменённый массив:

56 26 95 76

2) Пример 2.(Изменений не производится)

Исходная матрица из 3 строк и 3 столбцов( N=3):

10 11 -11

**B** = 4 10 -19

40 0 1

Исходный массив из 4 элементов(M=4):

C = -20 -50 -19 0

Q = 10+ 11 + 4 +10 + 40 + 1 = 76

=> **Q** <| **SummC** | => НЕ изменяем матрицу и массив

**SummC** = | -20 -50 - 19| = 89

**4. Таблица данных**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Описание (смысл),  диапазон, точность** | **Тип** | **Структура** | **Формат** |
| Входные данные | B | Заданная квадратная матрица B , |*bi j*|<=100 | цел | двумерный массив (5х5) | +XXX(:4) |
| ***n*** | Число строк и столбцов матрицы B  1 ≤ *n* ≤ 5 | цел | простая переменная | X(:1) |
| ***C*** | Заданный одномерный массив C , |*ci j*|<=100 | цел | одномерный массив (5) | +XXX(:4) |
| ***m*** | Число элементов массива C  1 ≤  ***m*** ≤ 5 | цел | простая переменная | X(:1) |
| Выходные данные | ***В*** | Изменённая матрица B ,  -99≤ *bi j*≤ 2600 | цел | двумерный массив (5х5) | +XXXX(:5) |
| ***С*** | Изменённый одномерный массив C ,  -99≤ *сi j*≤ 2600 | цел | Одномерный массив(5) | +XXXX(:5) |
| ***Q*** | Сумма положительных элементов матрицы В,  0≤***Q***≤2500 | цел | простая переменная | XXXX(:4) |
| ***SummC*** | Модуль суммы отрицательных элементов одномерного массива С,  0≤***SummC***≤500 | цел | простая переменная | +ХXX(:4) |
| Промежуточные данные | ***dat*** | Файл с исходными данными с именем вида Lab12\_in\_test\_<№>.txt | Текст. | Файл | --- |
| ***res*** | Файл с отчетом с именем вида Lab12\_out\_test\_<№>.txt | Текст. | Файл | --- |
| ***ii*** | Строка с некорректным элементом В[ii,jj]  1≤***ii***≤5 | цел | простая переменная | X(:1) |
| ***jj*** | Столбец с некорректным элементом В[ii,jj]  1≤ ***jj*** ≤5 | цел | простая переменная | X(:1) |
| ***Cii*** | Номер с некорр. элементом массива С[***с***ii]  1≤ ***с***ii ≤5 | цел | простая переменная | X(:1) |
| ***flagB*** | = True, если есть некорректные Вij, в противном случае False | лог | простая переменная | --- |
| ***flagC*** | = True, если есть некорректные Сi, в противном случае False | лог | простая переменная | --- |

**5.Входная форма** (Ввод осуществляется из файла Lab12\_in\_test\_<№>.txt)

<n>

<m>

<b[1,1]>...<b[1,n]>

. . . . . . . . .

<b[1,n]>...<b[n,n]>

<c[1]>...<c[m]>

**6.Выходная форма** (Вывод осуществляется в файл Lab12\_out\_test\_<№>.txt)

Программа с модулем

Исходная матрица из <n> строк и <n> столбцов:

<b[1,1]>...<b[1,n]>

. . . . . . . . .

<b[1,n]>...<b[n,n]>

Исходный массив из <m> элементов:

<c[1]>...<c[m]>

Некорректное число строк и столбцов матрицы.

Аном.1

Задача не решалась

Аном.2

Некорректное число элементов одномерного массива.

Задача не решалась

Некорректный элемент матрицы

Аном.3

B[<ii>,<jj>] = <B[ii,jj]>

Задача не решалась

Аном.4

Некорректный элемент массива

С[<cii>] = <C[cii]>

Задача не решалась

Сумма положительных элементов матрицы = <Q>

Сумма (по модулю) отрицательных элементов массива =<SummC>

Изменяем матрицу и массив

Изменённая матрица:

<b[1,1]>...<b[1,n]>

Сит.1

. . . . . . . . .

<b[1,n]>...<b[n,n]>

Изменённый массив:

<c[1]>...<c[m]>

Сит.2

Матрицу и массив не изменяем

**Имена входного и выходного файлов передаются как параметры программы**:

Первый – имя файла с исходными данными. Второй – имя файла для вывода исходных данных и результатов.

**7.Аномалии**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Описание | Условие возникновения | Реакция |
| 1 | Некорректное число строк и столбцов матрицы | (n<1)or(n>5) | Аном.1 |
| 2 | Некорректное число элементов массива | (m<1)or(m>5) | Аном.2 |
| 3 | Есть некорректный элемент в матрице | ƎiiƎjj(|B[ii,jj]|>100) | Аном.3 |
| 4 | Есть некорректный элемент в массиве | Ǝcii(|C[cii]|>100) | Аном.4 |

**8.** **Функциональные тесты**

|  |  |  |  |
| --- | --- | --- | --- |
| № Теста | Входные данные | Ожидаемый результат | Смысл теста |
| 1 | n=-1 | Аном.1 | Аномальная ситуация 1  n<1 |
| 2 | n=9 | Аном.1 | Аномальная ситуация 1 n>5 |
| 3 | m=0 | Аном.2 | Аномальная ситуация 2 m<1 |
| 4 | m=11 | Аном.2 | Аномальная ситуация 2 m>5 |
| 5 | n=2,m=3  101 -4  B = 100 5 C= (2 4 5) | Аном.3 B[1,1]=101 | Аномальная ситуация 3  B[1,1]>100 |
| 6 | n=4, m=5  10 -12 34 15  B = 0 0 -500 3 C= (-2 3 15 5 1)  1 2 4 5  6 7 8 8 | Аном.3 B[2,3]=-500 | Аномальная ситуация 3  B[2,3]<-100 |
| 7 | n=2, m=4  B = 10 11 C= (100 101 90 -1)  -5 90 | Аном.4  C[2]=101 | Аномальная ситуация 4  C[2]>100 |
| 8 | n=3, m=3  10 -5 6  B = -1 -2 -3 C= (-11 -21 -150)  4 5 6 | Аном.4  C[3]=-150 | Аномальная ситуация 4  C[3]<-100 |
| 9 | n=3, m=4    10 11 -11  B = 4 10 -19 C=( -20 -50 19 0)  40 0 1 | Q = 76 SummC=70  86 87 65  B = 80 86 57  116 76 77  C= (56 26 95 76) | Средние значения, происходит изменение матрицы и массива (сит.1), данные взяты из примера 1 |
| 10 | n=3, m=4    10 11 -11  B = 4 10 -19 C=( -20 -50 -19 0)  40 0 1 | Q = 76 SummC=89 | Средние значения, изменений массива и матрицы не происходит(сит.2), данные взяты из примера 2 |
| 11 | n=5, m=5  100 100 100 100 100  100 100 100 100 100  В = 100 100 100 100 100  100 100 100 100 100  100 100 100 100 100  С = (100 100 100 100 100) | Q=2500  SummC=0  2600... 2600  В = . . . . . . . . . . .  2600 ... 2600  С=(2600...2600) | Макс. нагрузка, в исходных данных взяты максимальные значения, причем в массиве С нет отрицательных элем., происходит изменение массива и матрицы(сит.1) |
| 12 | n=3, m=3  -12 -2 -4  В = -5 -6 -9 С=( 12 11 99)  -14 -1 -7 | Q=0  SummC=0 | В матрице нет положительных элементов, в массиве нет отрицательных элементов, изменений не происходит (сит.2) |
| 13 | n=3, m=3  -100 -100 -100  В = -100 -100 -100  -100 -100 -100    С = (-100 -100 -100) | Q=0  SummC=300 | В матрице и массиве все элементы минимальны и отрицательны, изменений не происходит (сит.2) |
| 14 | n=1, m=4  В = (75) С=( -11 -23 -4 100) | Q=75  SummC=38  В = (150)  С = (64 52 71 175) | Матрица состоит из одного положительного элемента, который больше SummC. Следовательно, изменения в матрице и массиве производятся.(сит.1) |
| 15 | n=1, m=4  В = (15) С = (-11 3 0 -5) | Q=0  SummC=16 | Матрица состоит из одного отрицательного элемента. Изменений нет(сит.2) |
| 16 | n =1, m=1  B= (19) C= (0) | Q=19  SummC=0  В = (38)  С = (19) | Матрица состоит из одного элемента, массив состоит из одного элемента. Происходят изменения(сит.1) |
| 17 | n =1, m=1  В = (-2) С = (-4) | Q=0  SummC=4 | Матрица состоит из одного элемента, массив состоит из одного элемента. Изменения не происходят (сит.2) |

**9. Метод**

Отделим ввод-вывод от обработки данных (собственно решения задачи). То есть, разделим нашу задачу на десять подзадач:

1. Подзадача А0.1. Ввести из файла и проверить значение количества строк и столбцов матрицы и количество элементов массива

2. Подзадача А0.2. Ввести матрицу из файла.

3. Подзадача А0.3. Ввести массив из файла.

4. Подзадача А0.4. Проверить значения элементов матрицы.

5. Подзадача А0.5. Проверить значения массива.

6. Подзадача А0.6. Найти сумму положительных элементов матрицы

7. Подзадача А0.7. Найти модуль суммы отрицательных элементов массива

8. Подзадача А0.8. Вывести матрицу

9. Подзадача А0.9. Вывести массив

10. Подзадача А0.10. Изменить массив и матрицу, увеличив каждый элемент на Q.

Сейчас, на нулевом уровне, опишем ввод-вывод простых переменных и массивов (А0.1, A0.2, А0.3,A0.8, A0.9,) и логику решения задачи в целом. Остальные пять задач оставим в виде абстракций.

Все подзадачи кроме первой выполним в виде процедур и функций, расположенных в отдельном модуле.

**10. Алгоритм**

Начало

Ввод n

+

Вывод по Аном.1

n<1 or n>5

**A0.1** -

Ввод m

m <1 or m >5

+

Вывод по Аном.2

-

Вх: n

Ввод матрицы из файла ReadMatrix(dat, n, B)

**A0.2**

Вх: n, B

Вых: B

Вывод матрицы в файл WriteMatrix(res, n, B)

**A0.8**

Вых: -

Вх: m

Ввод массива из файла ReadMassiv(dat,m,C)

**A0.3**

Вых: С

Вх: m, С

Вывод массива в файл WriteMassiv(res,m,C)

**A0.9**

Вх: n, B

Вых: -

Проверка элементов матрицы ProvMatrix(n, B, FlagB, ii, jj)

**А0.4**

Вых: ii,jj,FlagB

Вывод по Аном.3

FlagB

+

Вх: m, С

-

Проверка элементов маccива ProvMassiv(m,C,flagC,Cii)

**А0.5**

Вых: сii, FlagС

FlagС

+

Б

Вывод по Аном.4

-

А

Вх: n, B

Б

А

Подсчет суммы положительных элементов матрицы Q:=FindSummMatrix(n,B)

**А0.6**

Вх: m, С

Вых: Q

Подсчет модуля суммы отрицательных элементов массива

SummC:=abs(FindSummMassiv(m,C))

**А0.7**

Вых: SummC

Вывод Q, SummC

Q>SummC

-

Вывод сообщ. сит.2

+

Вх: m, С

Вых: B,С

Вых: -

Вх: n, B

Вх: n,m,B, С,Q

Вывод сообщ. сит.1

Конец

Вывод массива в файл WriteMassiv(res,m,C)

Вывод матрицы в файл WriteMatrix(res, n, B)

Изменение массива и матрице

IzmenMatrixMassiv(n,m,B,C,Q);

**А0.10**

**А0.8**

**А0.9**

Вых: -

**11. Программа на Delphi**

Имена входного и выходного файлов передаются как параметры программы: Первый – имя файла с исходными данными Второй – имя файла для вывода исходных данных и результатов

**Program** Lab12;

{$APPTYPE CONSOLE}

**uses** UnitLab12;

**var** {раздел описания переменных --------------------------------------------}

B: Matrix; C: Massiv;

n, m, ii, Cii, jj: shortint;

SummC,Q:integer;

flagB, flagC: Boolean;

dat, res: TextFile;

**begin** {раздел операторов ---------------------------------------------------}

assignFile(dat, ParamStr(1)); reset(dat);

assignFile(res, ParamStr(2)); rewrite(res);

writeln(res,' ':15, 'Программа с модулем');

{ввод n --------------------------------------------------------------------}

readln(dat, n);

**if** (N<1) **OR** (N>Nmax) **then**

**begin**

writeln(res,'Некорректное число строк и столбцов матрицы.'#13#10, ' Задача не решалась');

CloseFile(dat); CloseFile(res);

Halt;

**end**;

{ввод m --------------------------------------------------------------------}

readln(dat, m);

**if** (m<1) **OR** (m>Nmax) **then**

**begin**

writeln(res,'Некорректное число элементов одномерного массива.'#13#10, ' Задача не решалась');

CloseFile(dat); CloseFile(res);

Halt;

**end**;

{ввод и вывод матрицы B ----------------------------------------------------}

ReadMatrix(dat, n, B);

writeln(res,'Исходная матрица из ',n,' строк и ',n, ' столбцов:');

WriteMatrix(res, n, B);

{ввод и вывод массива С ----------------------------------------------------}

ReadMassiv(dat,m,C);

writeln(res,'Исходный массив из ',m,' элементов: ');

WriteMassiv(res,m,C);

CloseFile(dat);// ЗАКРЫВАЕМ ФАЙЛ НА ЧТЕНИЕ !!!

{ Проверка элементов матрицы ----------------------------------------------}

ProvMatrix(n, B, FlagB, ii, jj);

**If** FlagB **then**

**Begin**

Writeln(res,'Некорректный элемент матрицы');

Writeln(res,'B[', ii , ',' , jj , '] = ' , B[ii,jj]);

Writeln(res,'Задача не решалась');

CloseFile(res);

Halt;

**end**;

{ Проверка элементов маccива ----------------------------------------------}

ProvMassiv(m,C,flagC,Cii);

**If** FlagC **then**

**Begin**

Writeln(res,'Некорректный элемент массива ');

Writeln(res,'С[', Cii , '] = ' , C[Cii]);

Writeln(res,'Задача не решалась');

CloseFile(res);

Halt;

**end**;

{ Подсчет суммы положительных элементов матрицы и суммы отрицательных элементов массива

---------------------------------------------------------------------------

Q:=FindSummMatrix(n,B);

SummC:=abs(FindSummMassiv(m,C));

writeln(res,'Сумма положительных элементов матрицы = ',Q:2);

writeln(res,'Сумма (по модулю) отрицательных элементов массива =',summC:2);

**if** Q>(SummC) **then**

**begin**

Writeln(res,'Изменяем матрицу и массив');

IzmenMatrixMassiv(n,m,B,C,Q);

Writeln(res,'Изменённая матрица:');

WriteMatrix(res, n, B);

Writeln(res,'Изменённый массив:');

WriteMassiv(res,m,C);

**end**

**else**

**begin**

Writeln(res,'Матрицу и массив не изменяем');

**end**;

CloseFile(res);

**end**.

Модуль UnitLab12.pas cо всеми процедурами и функциями

**unit** UnitLab12;

**interface**

**const** nmax=5;

**type**

massiv=**array**[1..nmax] **of** integer;

**type**

matrix=**array**[1..nmax,1..nmax] **of** integer;

**Procedure** ReadMatrix(**var** dat: TextFile; **const** n: shortint; **var**{out} B: Matrix); {Ввод матрицы из файла, файл уже открыт}

**Procedure** WriteMatrix(**var** res: TextFile; **const** n: shortint; **var** B: Matrix); {Вывод матрицы в файл, файл уже открыт}

**Procedure** ReadMassiv(**var** dat: TextFile; **const** m: shortint; **var** {out} C: Massiv); {Ввод массива из файла, файл уже открыт}

**Procedure** WriteMassiv(**var** res: TextFile; **const** m: shortint; **var** C: Massiv); {Вывод массива в файл, файл уже открыт}

**Procedure** ProvMatrix(**const** n:shortint; **var** B: Matrix; **var** {out} FlagB: Boolean; **var** {out} ii, jj: shortint); // Проверка матрицы В

|  |  |  |
| --- | --- | --- |
| Таблица соответствия параметров | | |
| Формальные параметры | Фактические параметры | Тип |
| n | n | shortint |
| В | В | matrix |
| FlagB | FlagB | boolean |
| ii | ii | shortint |
| jj | jj | shortint |

**Procedure** ProvMassiv(**const** m:shortint; **var** C: massiv; **var** {out} FlagC: Boolean; **var**{out} Cii : shortint); //Проверка массива С

|  |  |  |
| --- | --- | --- |
| Таблица соответствия параметров | | |
| Формальные параметры | Фактические параметры | Тип |
| m | m | shortint |
| C | C | massiv |
| FlagC | FlagC | boolean |
| Cii | Cii | shortint |

**Function** FindSummMatrix(**const** n:shortint; **var** B: Matrix): integer; {Поиск суммы положительных элементов матрицы В}

|  |  |  |
| --- | --- | --- |
| Таблица соответствия параметров | | |
| Формальные параметры | Фактические параметры | Тип |
| n | n | shortint |
| В | В | matrix |
| FindSummMatrix(значение функции) | Q | integer |

**Function** FindSummMassiv(**const** m:shortint; **var** C: Massiv): integer; // Поиск суммы отрицательных элементов массива С

|  |  |  |
| --- | --- | --- |
| Таблица соответствия параметров | | |
| Формальные параметры | Фактические параметры | Тип |
| m | m | shortint |
| C | C | massiv |
| FindSummMassiv(значение функции) | SummC | integer |

**Procedure** IzmenMatrixMassiv(**const** n,m:shortint; {out} **var** B: Matrix; {out} **var** C:massiv; Q:integer); {Увеличение на Q значения каждого из элементов B и C}

|  |  |  |
| --- | --- | --- |
| Таблица соответствия параметров | | |
| Формальные параметры | Фактические параметры | Тип |
| n | n | shortint |
| В | В | matrix |
| m | m | shortint |
| C | C | massiv |
| Q | Q | integer |

**implementation**

{Ввод матрицы из файла, файл уже открыт }

**Procedure** ReadMatrix(**var** dat: TextFile; **const** n: shortint; **var**{out} B: Matrix);

**Var** i,j: byte;

**Begin**

**For** i:=1 **to** n **do**

**begin**

**For** j:=1 **to** n **do**

Read(dat, B[i,j]);

Readln(dat);

**end**;

**End**;

{Вывод матрицы в файл, файл уже открыт }

**Procedure** WriteMatrix(**var** res: TextFile; **const** n: shortint; **var** B: Matrix);

**Var** i,j: byte;

**Begin**

**For** i:=1 **to** n **do**

**begin**

**For** j:=1 **to** n **do**

Write(res, B[i,j]:3, ' ');

Writeln(res);

**end**;

**End**;

{Ввод массива из файла, файл уже открыт}

**Procedure** ReadMassiv(**var** dat: TextFile; **const** m: shortint; **var** {out} C: Massiv);

**Var** i: byte;

**Begin**

**For** i:=1 **to** m **do**

Read(dat, C[i]);

Readln(dat);

**End**;

{Вывод массива в файл, файл уже открыт}

**Procedure** WriteMassiv(**var** res: TextFile; **const** m: shortint; **var** C: Massiv);

**Var** i: byte;

**Begin**

**For** i:=1 **to** m **do**

Write(res, C[i], ' ');

Writeln(res);

**End**;

{Проверка матрицы В}

**Procedure** ProvMatrix(**const** n:shortint; **var** B: Matrix; **var** {out} FlagB: Boolean; **var** {out} ii, jj: shortint);

{Заглушка} writeln(’ Заглушка - Проверка элементов матрицы ’); {тест 5} // ii:=1; jj:=1; FlagВ:=true;

{тест 6} // ii:=2; jj:=3; FlagВ:=true;

{остальные тесты} FlagВ:=False;

{Проверка массива С}

**Procedure** ProvMassiv(**const** m:shortint; **var** C: massiv; **var** {out} FlagC: Boolean; **var**{out} Cii : shortint);

{Заглушка} writeln(’ Заглушка - Проверка элементов массива ’); {тест 7} // Сii:=2; FlagС:=true;

{тест 8} // Сii:=3; FlagС:=true;

{остальные тесты} FlagС:=False;

{Поиск суммы положительных элементов матрицы В}

**Function** FindSummMatrix(**const** n:shortint; **var** B: Matrix): integer;

**var**

i, j: byte;

Q:integer;

**Begin**

Q:=0;

**for** i:=1 **to** n **do**

**begin**

**for** j:=1 **to** n **do**

**if** B[i,j]>0 **then**

**begin**

Q:=Q+B[i,j];

**end**;

**end**;

FindSummMatrix:=Q;

**End**;

{ Поиск суммы отрицательных элементов массива С}

**Function** FindSummMassiv(**const** m:shortint; **var** C: Massiv): integer;

**var**

i: byte;

SummC:integer;

**Begin**

SummC:=0;

**for** i:=1 **to** m **do**

**if** c[i]<0 **then**

summC:=summC+c[i];

FindSummMassiv:=summC;

**End**;

{Увеличение на Q значения каждого из элементов B и C}

**Procedure** IzmenMatrixMassiv(**const** n,m:shortint; **var** B: Matrix; **var** C:massiv; Q:integer);

**var**

i, j: byte;

**Begin**

**for** i:=1 **to** n **do**

**for** j:=1 **to** n **do**

B[i,j]:=B[i,j]+Q;

**for** i:=1 **to** m **do**

C[i]:=C[i]+Q;

**End**;

**end**.

**Абстракция А0.4**

**1.Условие.**   
Проверить, есть ли в матрицы элементы, абсолютная величина которых больше 100   
**2.Уточненная постановка задачи.**Дана целочисленная квадратная матрица В из n строк и столбцов. Проверить (FlagВ), есть ли в матрицы элементы, абсолютная величина которых больше 100. Если есть, то присвоить FlagВ значение True и найти номер строки (ii) и столбца (jj) первого такого элемента, иначе присвоить FlagВ значение False.

**3.Примеры**

Тест 5. Есть некорректный элемент B[1,1]=101 (FlagB=True)

Тест 6. Есть некорректный элемент B[2,3]=-500 (FlagB=True)

Тест 7-17. Hекорректных элементов нет (FlagB=False)  
**4. Таблица данных**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Описание (смысл),  диапазон, точность** | **Тип** | **Структура** |
| Входные данные | **B** | Заданная квадратная матрица B ,  |*bi j*|<=100 | цел | двумерный массив (5х5) |
| ***n*** | Число строк и столбцов матрицы B  1 ≤ *n* ≤ 5 | цел | простая переменная |
| Выходные данные | ***ii*** | Строка с некорректным элементом В[ii,jj]  1≤***ii***≤5 | цел | простая переменная |
| ***jj*** | Столбец с некорректным элементом В[ii,jj]  1≤ ***jj*** ≤5 | цел | простая переменная |
| ***flagB*** | = True, если есть некорректные Вij, в противном случае False | лог | простая переменная |
| Промежуточные данные | ***i*** | Номер текущей строки,  1 ≤ *i* ≤ 5 | цел | простая переменная |
| ***j*** | Номер текущего столбца,  1 ≤ *j* ≤ 5 | цел | простая переменная |

**5.Входная форма.**

нет ввода/вывода

**6.Выходная форма**

нет ввода/вывода

**7. Аномалии**

нет ввода/вывода

**8. Тесты**Тест 5. Есть некорректный элемент B[1,1]=101 (FlagB=True)

Тест 6.Есть некорректный элемент B[2,3]=-500 (FlagB=True)

Тест 7-17. Hекорректных элементов нет (FlagB=False)

**9. Метод**

Истина, если есть в матрице элемент |Вi,j |>100;

Ложь, если нет такого элемента

Пусть FlagВ=

Предположим сначала, что такого элемента в матрице нет (FlagВ:=False). Затем будем просматривать строки, начиная с первой (i:=1) Пока не просмотрены все (i≤n) и не найден такой элемент (FlagВ=False)

В каждой строке будем просматривать элементы, начиная с первого (j:=1)

Пока не просмотрены все (j≤n) и не найден такой элемент (FlagВ=False)  
 Если рассматриваемый элемент Вij по модулю больше 100

То  
 Искомый элемент найден! (FlagВ:=True)   
 Запоминаем его местонахождение (ii:=i; jj:=j)  
 Переходим к следующему элементу в строке (j:=j+1)  
 Переходим к следующей строке

Найденные значения FlagВ, ii, jj будут искомыми.

**10. Алгоритм**

Вх: n, B

Начало А0.4

FlagB:=False

i:=1

i≤n and not flagB

-

j≤n and not flagB

j:=1

+

-

i:=i+1

+

Конец А0.4

|Bij|>100

+ -

j:=j+1

ii:=i; jj:=j; FlagВ:=True

Вых: ii,jj,FlagB

**11. Программный код. Процедура общего вида**

{Проверка матрицы В}

**Procedure** ProvMatrix(**const** n:shortint; **var** B: Matrix; **var** {out} FlagB: Boolean; **var** {out} ii, jj: shortint);

**var**

i, j: byte;

**Begin**

FlagB:=False; // Нет аномалий в матрице В

i:=1;

**while** (i<=n) **and not** flagB **do**

**begin**

j:=1;

**while** (j<=n) **and not** flagB **do**

**begin**

**if** abs(B[i,j]) > 100 **then**

**begin**

ii:=i; jj:=j; flagB:=True

**end**;

inc(j);

**end**;

inc(i);

**end**;

**End**;

**Абстракция А0.5**

**1.Условие.**   
Проверить, есть ли в массиве элементы, абсолютная величина которых больше 100   
**2.Уточненная постановка задачи.**Дан целочисленный одномерный массив С из m элементов. Проверить (FlagС), есть ли в массиве элементы, абсолютная величина которых больше 100. Если есть, то присвоить FlagС значение True и найти номер первого такого элемента (Сii), иначе присвоить FlagС значение False.

**3. Примеры**Тест 7. Есть некорректный элемент С[2]=101 (FlagС=True)

Тест 8. Есть некорректный элемент С[3]=-500 (FlagС=True)

Тест 9-17. Hекорректных элементов нет (FlagС=False)

**4.Таблица данных**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Описание (смысл),  диапазон, точность** | **Тип** | **Структура** |
| Входные данные | ***C*** | Заданный одномерный массив C , |*ci j*|<=100 | цел | одномерный массив (5) |
| ***m*** | Число элементов массива C  1 ≤  ***m*** ≤ 5 | цел | простая переменная |
| Выходные данные | ***Cii*** | Номер с некорр. элементом массива С[***с***ii]  1≤ ***с***ii ≤5 | цел | простая переменная |
| ***flagC*** | = True, если есть некорректные Сi, в противном случае False | лог | простая переменная |
| Промежуточные данные | ***i*** | Номер текущего элемента,  1 ≤ *i* ≤ 5 | цел | простая переменная |

**5.Входная форма.**

нет ввода/вывода

**6.Выходная форма**

нет ввода/вывода

**7. Аномалии**

нет ввода/вывода

**8. Тесты**Тест 7. Есть некорректный элемент С[2]=101 (FlagС=True)

Тест 8. Есть некорректный элемент С[3]=-500 (FlagС=True)

Тест 9-17. Hекорректных элементов нет (FlagС=False)

**9.Метод**

Истина, если есть в массиве элемент |Сi |>100;

Ложь, если нет такого элемента

Пусть FlagС=

Предположим сначала, что такого элемента в массиве нет (FlagС:=False). Затем будем просматривать элементы, начиная с первого (i:=1) Пока не просмотрены все (i≤n) и не найден такой элемент (FlagС=False)

Если рассматриваемый элемент Сi по модулю больше 100

То

Искомый элемент найден! (FlagС:=True) Запоминаем его местонахождение (Сii:=i)

Искомый элемент найден! (FlagВ:=True)   
   
 Переходим к следующему элементу (i:=i+1)

Найденные значения FlagС, Сii будут искомыми.

**10. Алгоритм**

Вх: m, С

Начало А0.5

FlagС:=False

i:=1

-

i≤m and not flagС

+

|Сi|>100

+ -

Сii:=i; FlagС:=True

i:=i+1

Конец А0.5

Вых: сii, FlagС

**11. Программный код(процедура общего вида)**

**Procedure** ProvMassiv(**const** m:shortint; **var** C: massiv; **var** {out} FlagC: Boolean; **var**{out} Cii : shortint);

**var**

i: byte;

**Begin**

FlagC:=False; // Нет аномалий в массиве С

i:=1;

**while** (i<=m) **and not** flagC **do**

**begin**

**if** abs(C[i]) > 100 **then**

**begin**

Cii:=i; flagC:=True

**end**;

inc(i);

**end**;

**End**;

**Абстракция А0.6**

**1.Условие**Найти сумму положительных элементов заданной матрицы В.

**2.Уточненная ПЗ**Дана квадратная целочисленная матрица В из n строк и n столбцов. Найти сумму Q положительных элементов матрицы

**3. Примеры**

См .пример 1 (тесты 9) в А0 – Среднее значение  
Тест 11 Максимальное значение Q=2500

Тест 12 Минимальное значение Q = 0 ( нет положительных элементов)

**4.Таблица данных**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Описание (смысл),  диапазон, точность** | **Тип** | **Структура** |
| Входные данные | **B** | Заданная квадратная матрица B ,  |*bi j*|<=100 | цел | двумерный массив (5х5) |
| ***n*** | Число строк и столбцов матрицы B  1 ≤ *n* ≤ 5 | цел | простая переменная |
| Выходные данные | ***Q*** | Сумма положительных элементов матрицы В,  0≤***Q***≤2500 | цел | простая переменная |
| Промежуточные данные | ***i*** | Номер текущей строки,  1 ≤ *i* ≤ 5 | цел | простая переменная |
| ***j*** | Номер текущего столбца,  1 ≤ *j* ≤ 5 | цел | простая переменная |

**5.Входная форма.**

нет ввода/вывода

**6.Выходная форма**

нет ввода/вывода

**7. Аномалии**

нет ввода/вывода

**8.Тесты**Тесты 9-17 (см. в А0 пункт 8. Функциональные тесты)

**9.Метод**

Пусть i – номер строки, а j – номер столбца текущего элемента двумерного массива B.

Примем за начальное значение суммы число 0 (Q:=0), т.к. Q=0+Q

Теперь по очереди с первой по последнюю строки (i=1..n):

Просматриваем с первого по последний элемент (j=1..n)

Если элемент положительный (B[i,j]>0)

ТО Увеличиваем сумму на значение этого элемента(Q:=Q+B[I,j])  
Полученное значение Q - искомое

Вх: n, B

Начало А0.6

**10.Алгоритм**

Q:=0;

i:=1;+1;n

j:=1;+1;n

Bij>0

- +

Q:=Q+ Bij;

Вых: Q

Конец А0.6

**11.Программный код(функция)**

{Поиск суммы положительных элементов матрицы В}

**Function** FindSummMatrix(**const** n:shortint; **var** B: Matrix): integer;

**var**

i, j: byte;

Q:integer;

**Begin**

Q:=0;

**for** i:=1 **to** n **do**

**begin**

**for** j:=1 **to** n **do**

**if** B[i,j]>0 **then**

**begin**

Q:=Q+B[i,j];

**end**;

**end**;

FindSummMatrix:=Q;

**End**;

**Абстракция А0.7**

**1.Условие**Найти модуль суммы отрицательных элементов заданного массива С, состоящего из m элементов.

**2.Уточненная ПЗ**Дан одномерный целочисленный массив С из m элементов. Найти сумму SummC отрицательных элементов массива. Затем взять абсолютную величину SummC.

**3. Примеры**

Те же, что и в А0

**4.Таблица данных**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Описание (смысл),  диапазон, точность** | **Тип** | **Структура** |
| Входные данные | ***C*** | Заданный одномерный массив C , |*ci j*|<=100 | цел | одномерный массив (5) |
| ***m*** | Число элементов массива C  1 ≤  ***m*** ≤ 5 | цел | простая переменная |
| Выходные данные | ***SummC*** | Сумма отрицательных элементов одномерного массива С,  -500≤***SummC***≤0 | цел | простая переменная |
| Промежуточные данные | ***i*** | Номер текущего элемента,  1 ≤ *i* ≤ 5 | цел | простая переменная |

**5.Входная форма.**

нет ввода/вывода

**6.Выходная форма**

нет ввода/вывода

**7. Аномалии**

нет ввода/вывода

**8. Тесты**

Тесты 9-17 в А0.

**9.Метод**

Пусть i – номер текущего элемента массива C.   
Примем за начальное значение суммы число 0 (SummC:=0), т.к. SummC=0+SummC

Теперь по очереди с первого по последний элемент (i=1..n):   
 Просматриваем элементы и   
 Если отрицателен (b[i]<0)  
 добавляем его к сумме (SummC= b[i]+SummC)  
Полученная сумма будет искомой.

Находим модуль(SummC:=| SummC|)

**10.Алгоритм**

Конец А0.7

SummC:=0;

Начало А0.7

i:=1;+1;n

Вх: m, С

Ci<0

+ -

Вых: SummC

SummC:=|SummC|;

SummC= С[i]+SummC

**11.Программный код (функция)**  
{ Поиск модуля суммы отрицательных элементов массива С}

**Function** FindSummMassiv(**const** m:shortint; **var** C: Massiv): integer;

**var**

i: byte;

SummC:integer;

**Begin**

SummC:=0;

**for** i:=1 **to** m **do**

**if** c[i]<0 **then**

summC:=summC+c[i];

FindSummMassiv:=abs(summC);

**End**;

**Абстракция А0.10**

**1.Условие**  
Увеличить на Q значения каждого из элементов матрицы В(n строк и n столбцов) и значения каждого из элементов одномерного массива С( m элементов)

**2. Уточненная ПЗ**

Задана квадратная целочисленная матрица В, в которой n строк и n столбцов. Дан одномерный целочисленный массив С, состоящий из m элементов. Изменить матрицу В, прибавив к каждому ее элементу(Bij) значение Q. Изменить массив С , прибавив к каждому его элементу(Сi) значение Q.

**3.Примеры**Примеры 1,2 из А0

4.Таблица данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Описание (смысл),  диапазон, точность** | **Тип** | **Структура** |
| Входные данные | **B** | Заданная квадратная матрица B , |*bi j*|<=100 | цел | двумерный массив (5х5) |
| ***n*** | Число строк и столбцов матрицы B  1 ≤ *n* ≤ 5 | цел | простая переменная |
| ***C*** | Заданный одномерный массив C , |*ci j*|<=100 | цел | одномерный массив (5) |
| ***m*** | Число элементов массива C  1 ≤  ***m*** ≤ 5 | цел | простая переменная |
| ***Q*** | Заданное положительное значение,  1≤***Q***≤2500 | цел | простая переменная |
| Выходные данные | ***В*** | Изменённая матрица B ,  -99≤ *bi j*≤ 2600 | цел | двумерный массив (5х5) |
| ***С*** | Изменённый одномерный массив C ,  -99≤ *сi j*≤ 2600 | цел | Одномерный массив(5) |
| Промежуточные данные | ***i*** | Номер текущей строки(или номер текущего элемента в случае с массивом),  1 ≤ *i* ≤ 5 | цел | простая переменная |
| ***j*** | Номер текущего столбца,  1 ≤ *j* ≤ 5 | цел | простая переменная |

**5.Входная форма.**

нет ввода/вывода

**6.Выходная форма**

нет ввода/вывода

**7. Аномалии**

нет ввода/вывода

**8. Тесты**

Тесты 9,11,14,16 в А0

**9.Метод**

Пусть i – номер строки, а j – номер столбца текущего элемента матрицы B.

Теперь по очереди с первой по последнюю строки (i=1..n):  
 Просматриваем с первого по последний элемент (j=1..n)

Добавляем к значению текущего элемента значение Q (Bij=Bij+Q)

Изменённая матрица В будет искомой.  
  
Теперь пусть i – номер текущего элемента массива С  
 По очереди с первого по последний элемент (i=1..m):  
 Просматриваем элементы и

Прибавляем к каждому значение Q ( Ci:=Ci+Q)

Изменённый массив С будет искомым.

**10. Алгоритм**

Вх: n,m,B, С,Q

Начало А0.10

i:=1;+1;n

j:=1;+1;n

В[ij]:=Q+B[ij];

i:=1;+1;n

C[i]:=Q+C[i];

Конец А0.10

Вых: B,С

**11.Программный код**

{Увеличение на Q значения каждого из элементов B и C}

**Procedure** IzmenMatrixMassiv(**const** n,m:shortint; **var** B: Matrix; **var** C:massiv; Q:integer);

**var**

i, j: byte;

**Begin**

**for** i:=1 **to** n **do**

**for** j:=1 **to** n **do**

B[i,j]:=B[i,j]+Q;

**for** i:=1 **to** m **do**

C[i]:=C[i]+Q;

**End**;

**СТРУКТУРНЫЕ ТЕСТЫ**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Структура** | **Тест№** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** |
| If (N<1) **OR** (N>5) then | |  | | | | | | | | | | | | | | | | |
| N<1 истина  N>5 истина | | Не возможно одновременное выполнение этих условий | | | | | | | | | | | | | | | | |
| N<1 истина  N>5 ложь (N=5) | | Не возможно одновременное выполнение этих условий | | | | | | | | | | | | | | | | |
| N<1 истина  N>5 ложь (N<5) | | n = -1 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| N<1 ложь (N=1)  N>5 истина | | Не возможно одновременное выполнение этих условий | | | | | | | | | | | | | | | | |
| N<1 ложь(N>1)  N>5 истина | | - | n=9 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | **-** |
| N<1 ложь (N=1)  N>5 ложь(N=5) | | Не возможно одновременное выполнение этих условий | | | | | | | | | | | | | | | | |
| N<1 ложь(N>1)  N>5 ложь(N=5) | |  |  |  |  |  |  |  |  |  |  | n=5 |  |  |  |  |  |  |
| N<1 ложь (N=1)  N>5 ложь (N<5) | |  |  |  |  |  |  |  |  |  |  |  |  |  | n=1 | n=1 | n=1 | n=1 |
| N<1 ложь(N>1)  N>5 ложь (N<5) | |  |  |  |  | n=2 | n=4 | n=2 | n=3 | n=3 | n=3 |  | n=3 | n=3 |  |  |  |  |
| If (M<1) **OR** (M>5) | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| M<1 истина  M>5 истина | | Не возможно одновременное выполнение этих условий | | | | | | | | | | | | | | | | |
| M<1 истина  M>5 ложь (M=5) | | Не возможно одновременное выполнение этих условий | | | | | | | | | | | | | | | | |
| M<1 истина  M>5 ложь (N<5) | | m=0 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | **-** | **-** |
| M<1 ложь (M=1)  M>5 истина | | Не возможно одновременное выполнение этих условий | | | | | | | | | | | | | | | | |
| M<1 ложь(M>1)  M>5 истина | | - | m=11 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | **-** |
| M<1 ложь (M=1)  M>5 ложь(M=5) | | Не возможно одновременное выполнение этих условий | | | | | | | | | | | | | | | | |
| M<1 ложь(M>1)  M>5 ложь(M=5) | |  |  |  |  |  |  | 5 |  |  |  | 5 |  |  |  |  |  |  |
| M<1 ложь (M=1)  M>5 ложь (M<5) | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 | 1 |
| M<1 ложь(M>1)  M>5 ложь (M<5) | |  |  |  |  | 3 |  | 4 | 3 | 4 | 4 |  |  |  |  |  |  |  |
| **Ввод/вывод матрицы** For i:=1 to n do | | Н  Е  Т  В  В  О  Д  А  И  П  Р  О  В  Е  Р  К  И  М  А  Т  Р  И  Ц  Ы | Н  Е  Т  В  В  О  Д  А  И  П  Р  О  В  Е  Р  К  И  М  А  Т  Р  И  Ц  Ы | Н  Е  Т  В  В  О  Д  А  И  П  Р  О  В  Е  Р  К  И  М  А  Т  Р  И  Ц  Ы | Н  Е  Т  В  В  О  Д  А  И  П  Р  О  В  Е  Р  К  И  М  А  Т  Р  И  Ц  Ы |  | | | | | | | | | | | | |
| 0 раз (i>n) | | Не возможно, n≥1 | | | | | | | | | | | | |
| 1 раз (i=n) | |  |  |  |  |  |  |  |  |  | 1 | 1 | 1 | 1 |
| более 1 раза (i<n) | | 2 | 4 | 2 | 3 | 3 | 3 |  | 3 |  |  |  |  |  |
| For j:=1 to n do | |  | | | | | | | | | | | | |
| 0 раз | | Не возможно, n≥1 | | | | | | | | | | | | |
| 1 раз | |  |  |  |  |  |  |  |  |  | 1 | 1 | 1 | 1 |
| более 1 раза | | 2 | 4 | 2 | 3 | 3 | 3 | 5 | 3 |  |  |  |  |  |
| **Проверка элементов матрицы**  while (i<=n) and not flagB do | |  | | | | | | | | | | | | |
| (i<=n) истина (i<n)  flagB ложь | | - | + | + | + | + | + | + | + | + | - | - | - | - |
| (i<=n) истина (i=n)  flagB ложь | | - | - | + | + | + | + | + | + | + | + | + | + | + |
| (i<=n) истина (i<n)  flagB истина | | + | + | - | - | - | - | - | - | - | - | - | - | - |
| (i<=n) истина (i=n)  flagB истина | | - | - | - | - | - | - | - | - | - | - | - | - | - |
| (i<=n) ложь (i>n) flagB ложь | | - | - | + | + | + | + | + | + | + | - | - | - | - |
| (i<=n) ложь (i>n) flagB истина | | + | - | - | - | - | - | - | - | - | - | - | - | - |
| while (j<=n) and not flagB do | |  | | | | | | | | | | | | |
| (j<=n) истина (j<n) flagB ложь | | - | + | + | + | + | + | + | + | + | + | + | + | + |
| (j<=n) истина (j=n) flagB ложь | | - | - | + | + | + | + | + | + | + | + | + | + | + |
| (j<=n) истина (j<n) flagB истина | | - | - | - | - | - | - | - | - | - | - | - | - | - |
| (j<=n) истина (j=n) flagB истина | | - | + | - | - | - | - | - | - | - | - | - | - | - |
| (j<=n) ложь (j>n) flagB ложь | | - | - | + | + | + | + | + | + | + | + | + | + | + |
| (j<=n) ложь (j>n) flagB истина | | + | - | - | - | - | - | - | - | - | - | - | - | - |
| if abs(B[i,j]) > 100 then | |  | | | | | | | | | | | | |
| abs(B[i,j]) > 100 | | + | + | - | - | - | - | - | - | - | - | - | - | - |
| abs(B[i,j]) < 100 | | - | - | + | + | + | + | + | + | + | + | + | + | + |
| abs(B[i,j]) = 100 | | - | - | - | - | - | - | + | - | + | - | - | - | - |
| If FlagB then | |  | | | | | | | | | | | | |
| FlagB истина | | + | + | - | - | - | - | - | - | - | - | - | - | - |
| FlagBложь | | - | - | + | + | + | + | + | + | + | + | + | + | + |
| if abs(С[i]) > 100 then | |  | | | | | | | | | | | | |
| abs(С[i])> 100 | | - | - | + | + | - | - | - | - | - | - | - | - | - |
| abs(С[i])< 100 | | + | + | - | - | + | + | - | + | + | + | + | + | + |
| abs(С[i]]) = 100 | | - | - | - | - | - | - | + | - | - | - | - | - | - |
| **Проверка элементов массива** If FlagС then | |  | | | | | | | | | | | | |
| FlagС истина | | - | - | + | + | - | - | - | - | - | - | - | - | - |
| FlagСложь | | + | + | - | - | - | - | - | - | - | - | - | - | - |
| **Сумма в матрице**  For i:=1 to n do | | З  А  Д  А  Ч  А  Н  Е  Р  Е  Ш  А  Л  А  С  Ь | З  А  Д  А  Ч  А  Н  Е  Р  Е  Ш  А  Л  А  С  Ь | З  А  Д  А  Ч  А  Н  Е  Р  Е  Ш  А  Л  А  С  Ь | З  А  Д  А  Ч  А  Н  Е  Р  Е  Ш  А  Л  А  С  Ь |  | | | | | | | | |
| 0 раз | | Не возможно( далее обозначаем «\*») | | | | | | | | |
| 1 раз | | - | - | - | - | - | + | + | + | + |
| Более 1 раза | | + | + | + | + | + | - | - | - | - |
| For j:=1 to n do | |  | | | | | | | | |
| 0 раз | | Не возможно\* | | | | | | | | |
| 1 раз | | - | - | - | - | - | + | + | + | + |
| Более 1 раза | | + | + | + | + | + | - | - | - | - |
| **if** B[i,j]>0 **then** | |  |  |  |  |  |  |  |  |  |
| B[i,j]>0 | | + | + | + | - | - | + | + | + | - |
| B[i,j]<0 | | + | + | - | + | + | - | - | - | + |
| B[i,j]=0 | | + | + | - | - | - | - | - | - | - |
| **Сумма в массиве** For i:=1 to n do | |  | | | | | | | | |
| 0 раз | | Не возможно\* | | | | | | | | |
| 1 раз | | - | - | - | - | - | - | - | + | + |
| Более 1 раза | | + | + | + | + | + | + | + | - | - |
| **if** С[i]<0 **then** | |  | | | | | | | | |
| С[i]<0 | | + | + | - | - | + | + | + | + | + |
| С[i]>0 | | + | + | + | + | - | + | + | + | - |
| С[i]=0 | | + | + | - | - | - | - | - | + | - |
| **Изменение матрицы**  For i:=1 to n do | |  |  |  |  |  |  |  |  |  |
| 0 раз | | \* | З  а  д  а  ч  а  не  р е  а л а с ь | \* | З  а  д  а  ч  а  не  р е  а л а с ь | З  а  д  а  ч  а  не  р е  а л а с ь | \* | З  а  д  а  ч  а  не  р е  а л а с ь | \* | З  а  д  а  ч  а  не  р е  а л а с ь |
| 1 раз | | - | - | + | + |
| Более 1 раза | | + | + | - | - |
| For j:=1 to n do | |  |  |  |  |
| 0 раз | | \* | \* | \* | \* |
| 1 раз | | - | - | + | + |
| Более 1 раза | | + | + | - | - |
| **Изменение массива**  For i:=1 to n do | |  |  |  |  |
| 0 раз | | \* | \* | \* | \* |
| 1 раз | | - | - | - | + |
| Более 1 раза | | + | + | + | - |