

## Практическое занятие 6. Процедуры и параметры

**Задание:** 1) синтаксис описания и вызова процедур с параметрами; 2) оформление подзадач в виде процедур - примеры; 3) прокомментировать пример задачи «Три массива» с процедурами.

### Пример решения задачи «Три массива»: Спецификация с комментариями

**Задача «Три массива» из Лабораторной работы №11. Абстракция A0.**

#### 1. Постановка задачи (ПЗ)

**Задание:** Написать *две программы* обработки трех двумерных массивов в соответствии с условием. В первой версии программы использовать только процедуры общего вида, не менее двух. Во второй версии программы должна быть хотя бы одна функция.

**Условие:** Даны три двумерных массива:  $A (na \times na)$ ,  $B (nb \times nb)$  и  $C (nc \times nc)$ . Для каждого массива  $A, B, C$  решить задачу из Лаб. работы №8 и определить в каких и скольких из трех заданных массивов есть упорядоченные столбцы. Оформить в виде процедур ввод и вывод матрицы, решение задачи ЛР8, вывод созданного массива и, по желанию, поиск количества массивов. Размеры и значения элементов массивов вводить из одного общего или трех разных текстовых файлов, результат и исходные данные также вывести в пользовательский текстовый файл.

**Задача из Лабораторной работы №8:**

**Задание:** Дана матрица  $X$ , состоящая из  $n$  строк и  $n$  столбцов. С использованием единственного (не учитывая ввод и вывод) двукратного цикла решить задачу со следующим условием.

**Условие:** Получить массив  $Y$  по правилу:  $Y_j$  получает значение *TRUE*, если элементы  $j$ -го столбца упорядочены по убыванию своих значений, иначе  $Y_j$  получает значение *FALSE*. Подсчитать также количество ( $kol$ ) упорядоченных по убыванию столбцов в матрице.

#### 2. Уточненная ПЗ

**Условие:** Даны три целочисленных двумерных массива:  $A (na \times na)$ ,  $B (nb \times nb)$  и  $C (nc \times nc)$ . Для каждого массива  $A, B, C$  получить массив  $Y$  по правилу:  $Y_j$  получает значение *TRUE*, если элементы  $j$ -го столбца упорядочены по убыванию своих значений, иначе  $Y_j$  получает значение *FALSE*; и подсчитать также количество ( $kol[A]$ ,  $kol[B]$ ,  $kol[C]$ ) упорядоченных по убыванию столбцов в матрице. Затем определить в каких ( $S$ ) и скольких ( $k$ ) из трех заданных массивов есть упорядоченные столбцы. В случае отсутствия упорядоченных столбцов во всех массивах, выдать сообщение об этом.

Оформить в виде процедур ввод и вывод матрицы, решение задачи 2.5.2, вывод созданного массива и поиск количества массивов с упорядоченными столбцами. Для решения подзадачи создания массива  $Y$  и подсчета количества упорядоченных столбцов в отдельной матрице использовать а) процедуру общего вида; б) функцию.

Размеры и значения элементов массивов вводить из трех разных текстовых файлов, результат и исходные данные также вывести в пользовательский текстовый файл.

Размеры массивов считать лежащими в диапазоне от 2 до 10. Аномальные ситуации не рассматривать, считать, что все исходные данные корректны.

Упорядоченными по убыванию считать столбцы, упорядоченные строго по убыванию, т.е. каждый следующий элемент столбца должен быть строго меньше предыдущего.

#### 3. Пример

Пусть заданы три матрицы:

$na=2$ $A (2 \times 2)$ <table><tr><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td></tr></table>	1	1	0	1	$nb=3$ $B (3 \times 3)$ <table><tr><td>1</td><td>2</td><td>5</td></tr><tr><td>0</td><td>1</td><td>4</td></tr><tr><td>1</td><td>0</td><td>-3</td></tr></table>	1	2	5	0	1	4	1	0	-3	$nc=4$ $C (4 \times 4)$ <table><tr><td>100</td><td>100</td><td>-100</td><td>0</td></tr><tr><td>0</td><td>100</td><td>-100</td><td>0</td></tr><tr><td>-100</td><td>100</td><td>-100</td><td>0</td></tr><tr><td>-100</td><td>100</td><td>-100</td><td>0</td></tr></table>	100	100	-100	0	0	100	-100	0	-100	100	-100	0	-100	100	-100	0
1	1																														
0	1																														
1	2	5																													
0	1	4																													
1	0	-3																													
100	100	-100	0																												
0	100	-100	0																												
-100	100	-100	0																												
-100	100	-100	0																												
Один упорядоченный столбец – первый $kol['A']=1$ $Y=(True, False)$	Два упорядоченных столбца – второй и третий $kol['B']=2$ $Y=(False, True, True)$	Нет столбцов с упорядоченными по убыванию элементами $kol['C']=0$ $Y=(False, False, False, False)$																													
Массивов с упорядоченными столбцами два: $k=2$ , $S='A,B'$																															

## 4. Таблица данных

Класс	Имя	Описание (смысл), диапазон, точность	Тип	Структура	Формат
Входные данные	<i>na</i>	число строк и столбцов в A, $2 \leq na \leq 10$	цел	простая переменная	XX (:2)
	<i>A</i>	Первый массив $ A_{ij}  \leq 100$	цел	двухмерный массив (10x10)	+XXX (:4)
	<i>nb</i>	число строк и столбцов в A, $2 \leq nb \leq 10$	цел	простая переменная	XX (:2)
	<i>B</i>	Второй массив $ B_{ij}  \leq 100$	цел	двухмерный массив (10x10)	+XXX (:4)
	<i>nc</i>	число строк и столбцов в A, $2 \leq nc \leq 10$	цел	простая переменная	XX (:2)
	<i>C</i>	Третий массив $ C_{ij}  \leq 100$	цел	двухмерный массив (10x10)	+XXX (:4)
Выходные данные	<i>Y</i>	Создаваемый трижды массив	лог	одномерный массив(10)	TRUE/ FALSE (:5)
	<i>kol</i>	Количество упорядоченных столбцов в A,B,C, $0 \leq kol_{ch} \leq 10$ , Индексы – символы <i>ch</i> : 'A'..'C'	цел	одномерный массив(3)	XX (:2)
	<i>k</i>	Количество массивов с упорядоченными столбцами, $0 \leq k \leq 3$	цел	простая переменная	XX (:2)
	<i>S</i>	Перечисление имен массивов с упорядоченными столбцами	символ	Строка[5]	'', 'A', 'B,C', ..., 'A,B,C'

Промежуточные данные*	<i>res</i>	Файл для вывода исходных данных и результатов, <i>mas3res</i> <№>.txt			---
	<i>InOk</i>	=True, если не найдено аномалий при вводе, иначе False			---
	<i>OutOk</i>	=True, если не найдено аномалий при открытии выходного файла, иначе False			---
					---

\*Промежуточные данные заполняются не сразу, а по мере необходимости использовать дополнительные переменные

5. Входная форма (3 текстовых файла с именами *massiv*<№>.txt одинакового формата):

Первые три параметра программы – имена файлов с размерами и значениями массивов A,B,C.

Обр1

```
<nx>
<X[ 1, 1]>...<X[ 1,nx]>
```

Обр2

```
. . . . .
<X[nx, 1]>...<X[nx,nx]>
```

Здесь для первой матрицы *nx* соответствует *na*, *X* соответствует A, для второй матрицы *nx* соответствует *nb*, *X* соответствует B, для третьей матрицы *nx* соответствует *nc*, *X* соответствует C. См. процедуру *Vvod* – абстракция A0.1

## 6. Выходная форма: (имя файла *mas3res<№>.txt* – четвертый параметр программы)

Обр0	<15 пробелов> Три массива. Вариант № 0
Обр3a	Задан массив <'A'> (<na>x<na>) : <A[ 1, 1]>...<A[ 1,na]> ... <A[na, 1]>...<A[na,na]>
Обр3b	Задан массив <'B'> (<nb>x<nb>) : <B[ 1, 1]>...<B[ 1,nb]> ... <B[nb, 1]>...<B[nb,nb]>
Обр3c	Задан массив <'C'> (<nc>x<nc>) : <C[ 1, 1]>...<C[ 1,nc]> ... <C[nc, 1]>...<C[nc,nc]>
Обр4.1a	На основе массива <'A'> получен массив Y: <Y[1]>...<Y[na]>
Обр4.2a	kol<'A'> = <kol['A']>
Обр4.1b	На основе массива <'B'> получен массив Y: <Y[1]>...<Y[nb]>
Обр4.2b	kol<'B'> = <kol['B']>
Обр4.1c	На основе массива <'C'> получен массив Y: <Y[1]>...<Y[nc]>
Обр4.2c	kol<'C'> = <kol['C']>
Обр5	Всего найдено <k> массивов с упорядоченными столбцами: <S>
Обр6	Не найдено ни одного массива с упорядоченными столбцами

## 7. Аномалии

Рассмотрим аномальные ситуации, возникающие при работе с файлами.

Некорректные значения входных данных не рассматриваем, но по желанию можете сделать ( $na < 2$ ,  $na > 10$ ,  $nb < 2$ ,  $nb > 10$ ,  $nc < 2$ ,  $nc > 10$ ,  $\exists i \exists j: |A[i,j]| > 100$ ,  $\exists i \exists j: |B[i,j]| > 100$ ,  $\exists i \exists j: |C[i,j]| > 100$ ,).

№	Описание	Условие возникновения	Реакция на аномалию
1	Программе передано менее 4 параметров	$ParamCount < 4$	Сообщение: «Мало параметров» Действие: Завершение работы программы
2	Невозможно открыть файл для ввода массива	Нет файла или не хватает прав для его просмотра	Сообщение: «Невозможно открыть файл <имя файла>» Действие: $InOk := False$ , затем завершение работы программы
3	Невозможно открыть файл для вывода исходных данных и результата	Уже есть такой файл и он доступен только для чтения, или не хватает прав для его создания или записи	Сообщение: «Невозможно создать файл <имя файла>» Действие: $InOut := False$ , затем завершение работы программы
4	Ошибка при чтении из файла	Формат файла не соответствует форме ввода	Сообщение: «Некорректный формат файла <имя файла>» Действие: $InOk := False$ , затем завершение работы программы
5	Ошибка при записи в файл	Закончилось место или отсоединен съемный диск	Не рассматриваем Аварийное завершение

## 8. Функциональные тесты

Исходные данные								Результаты		Тест№
	аном	граница	сред		сред	граница	аном	Kol <sub>A</sub>	макс =10	5,8
Na	<2	2	( 3	,	9 )	10	>10		мин = 0	3,4,6,7
Тест№	---	1,2,4,6,7	3	←	←	5,8	---		сред = (0,10)	1,2
	аном	граница	сред		сред	граница	аном		не суц = не возм.	---
A[i,j]	<-100	-100	[-99,0)	0	(0,99]	100	>100		0 = см. мин	---
Тест№	---	3	4,6,7	1 3	1-2, 4-8	3	---		Макс.нагрузка = 10 упор.столбцов	5,8
	аном	граница	сред		сред	граница	аном	kol <sub>B</sub>	макс =10	5,7
Nb	<2	2	( 3	,	9 )	10	>10		мин = 0	2,4,6,8
Тест№	---	4,6,8	1-3	←	←	5,7	---		сред = (0,10)	1,3
	аном	граница	сред		сред	граница	аном		не суц = не возм.	---
B[i,j]	<-100	-100	[-99,0)	0	(0,99]	100	>100		0 = см. мин	---
Тест№	---	2	1,3,4,6, 8	1 3	1,3-8	2	---		Макс.нагрузка = 10 упор.столбцов	5,7
	аном	граница	сред		сред	граница	аном	kol <sub>C</sub>	макс =10	5,6
Nc	<2	2	( 3	,	9 )	10	>10		мин = 0	1,4,7,8
Тест№	---	3-4,7-8	1-2	←	←	5-6	---		сред = (0,10)	2,3
	аном	граница	сред		сред	граница	аном		не суц = не возм.	---
C[i,j]	<-100	-100	[-99,0)	0	(0,99]	100	>100		0 = см. мин	---
Тест№	---	1	2,4,7-8	1 3	2-8	1	---		Макс.нагрузка = 10 упор.столбцов	5,6
<p>Поскольку ввод и решение задачи 8 для трёх матриц будет осуществляться <i>тремя вызовами одной</i> процедуры ввода, <i>тремя вызовами одной</i> процедуры вывода и <i>тремя вызовами одной</i> процедуры решения задачи 8, то для проверки их кода достаточно, чтобы тесты с граничными и пограничными значениями (<i>na</i>, <i>nb</i>, <i>nc</i>, <i>A<sub>ij</sub></i>, <i>B<sub>ij</sub></i>, <i>C<sub>ij</sub></i>, <i>kol<sub>ch</sub></i>, <i>Y</i>) были выполнены для хотя бы одного из трёх массивов.</p> <p>Но этого может оказаться недостаточно для проверки алгоритма новой задачи – поиска <i>k</i> и <i>S</i>.</p>								Y	макс = все True	5,(6-8)
									мин = все False	4,...
									сред = есть и True и False	кроме 4-5
									не суц = не возм	---
									0 = см. все False	---
									Макс.нагрузка = все 10 упорядочены	5
Общие результаты										
K	макс =3				5		S	макс = 'A,B,C'		5
	мин = 0				4			мин = ''		4
	сред = (0,3)				1-3, 6-8			сред = 'A','B','C', 'A,B', 'A,C', 'B,C'		6-8, 1-3
	не суц = не возм.				---			не суц = не возм.		---
	0 = см. мин				---			0 = см. мин		---
	Макс.нагрузка = все 0, все kol <sub>ch</sub> >0				4-5			Макс.нагрузка = 'A,B,C'		5

## Функциональные тесты

Тест №	Входные данные						Ожидаемый результат	Смысл теста																														
1	$na=2, A (2 \times 2)$ <table><tr><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td></tr></table>		1	1	0	1	$nb=3, B (3 \times 3)$ <table><tr><td>1</td><td>2</td><td>5</td></tr><tr><td>0</td><td>1</td><td>4</td></tr><tr><td>1</td><td>0</td><td>-3</td></tr></table>		1	2	5	0	1	4	1	0	-3	$nc=4, C (4 \times 4)$ <table><tr><td>100</td><td>100</td><td>-100</td><td>0</td></tr><tr><td>0</td><td>100</td><td>-100</td><td>0</td></tr><tr><td>-100</td><td>100</td><td>-100</td><td>0</td></tr><tr><td>-100</td><td>100</td><td>-100</td><td>0</td></tr></table>		100	100	-100	0	0	100	-100	0	-100	100	-100	0	-100	100	-100	0	$Y=(T,F) Kol_A=1$ $Y=(F,T,T) Kol_B=2$ $Y=(F...F) Kol_C=0$ $k=2$ $S='A,B'$ (обр7)	Есть два массива (A,B) с упорядоченными столбцами; данные взяты из примера	
1	1																																					
0	1																																					
1	2	5																																				
0	1	4																																				
1	0	-3																																				
100	100	-100	0																																			
0	100	-100	0																																			
-100	100	-100	0																																			
-100	100	-100	0																																			
2	$na=2, A (2 \times 2)$ <table><tr><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td></tr></table>		1	1	0	1	$nb=4, B (4 \times 4)$ <table><tr><td>100</td><td>100</td><td>-100</td><td>0</td></tr><tr><td>0</td><td>100</td><td>-100</td><td>0</td></tr><tr><td>-100</td><td>100</td><td>-100</td><td>0</td></tr><tr><td>-100</td><td>100</td><td>-100</td><td>0</td></tr></table>		100	100	-100	0	0	100	-100	0	-100	100	-100	0	-100	100	-100	0	$nc=3, C (3 \times 3)$ <table><tr><td>1</td><td>2</td><td>5</td></tr><tr><td>0</td><td>1</td><td>4</td></tr><tr><td>1</td><td>0</td><td>-3</td></tr></table>		1	2	5	0	1	4	1	0	-3	$Y=(T,F) Kol_A=1$ $Y=(F...F) Kol_B=0$ $Y=(F,T,T) Kol_C=2$ $k=2$ $S='A,C'$ (обр7)	Есть два массива (A,C) с упорядоченными столбцами; данные взяты из примера, но в другом порядке	
1	1																																					
0	1																																					
100	100	-100	0																																			
0	100	-100	0																																			
-100	100	-100	0																																			
-100	100	-100	0																																			
1	2	5																																				
0	1	4																																				
1	0	-3																																				
3	$na=4, A (4 \times 4)$ <table><tr><td>100</td><td>100</td><td>-100</td><td>0</td></tr><tr><td>0</td><td>100</td><td>-100</td><td>0</td></tr><tr><td>-100</td><td>100</td><td>-100</td><td>0</td></tr><tr><td>-100</td><td>100</td><td>-100</td><td>0</td></tr></table>			100	100	-100	0	0	100	-100	0	-100	100	-100	0	-100	100	-100	0	$nb=3, B (3 \times 3)$ <table><tr><td>1</td><td>2</td><td>5</td></tr><tr><td>0</td><td>1</td><td>4</td></tr><tr><td>1</td><td>0</td><td>-3</td></tr></table>		1	2	5	0	1	4	1	0	-3	$nc=2, C (2 \times 2)$ <table><tr><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td></tr></table>		1	1	0	1	$Y=(F...F) Kol_A=0$ $Y=(F,T,T) Kol_B=2$ $Y=(T,F) Kol_C=1$ $k=2$ $S='B,C'$ (обр7)	Есть два массива (B,C) с упорядоченными столбцами; данные взяты из примера, но в другом порядке
100	100	-100	0																																			
0	100	-100	0																																			
-100	100	-100	0																																			
-100	100	-100	0																																			
1	2	5																																				
0	1	4																																				
1	0	-3																																				
1	1																																					
0	1																																					
4	$na=2, A (2 \times 2)$ <table><tr><td>10</td><td>-5</td></tr><tr><td>50</td><td>10</td></tr></table>		10	-5	50	10	$nb=2, B (2 \times 2)$ <table><tr><td>10</td><td>-5</td></tr><tr><td>50</td><td>10</td></tr></table>		10	-5	50	10	$nc=2, C (2 \times 2)$ <table><tr><td>10</td><td>-5</td></tr><tr><td>50</td><td>10</td></tr></table>		10	-5	50	10	$Y=(F,F) Kol_A=0$ $Y=(F,F) Kol_B=0$ $Y=(F,F) Kol_C=0$ $k=0$ $S=""$ (обр8)	Нет ни одного упорядоченного столбца; Минимальные размеры Данные из одного файла																		
10	-5																																					
50	10																																					
10	-5																																					
50	10																																					
10	-5																																					
50	10																																					
5	$na=10, A (10 \times 10)$ <table><tr><td>10</td><td>...</td><td>10</td></tr><tr><td>...</td><td>...</td><td>...</td></tr><tr><td>1</td><td>...</td><td>1</td></tr></table>		10	...	10	...	...	...	1	...	1	$nb=10, B (10 \times 10)$ <table><tr><td>10</td><td>...</td><td>10</td></tr><tr><td>...</td><td>...</td><td>...</td></tr><tr><td>1</td><td>...</td><td>1</td></tr></table>		10	...	10	...	...	...	1	...	1	$nc=10, C (10 \times 10)$ <table><tr><td>10</td><td>...</td><td>10</td></tr><tr><td>...</td><td>...</td><td>...</td></tr><tr><td>1</td><td>...</td><td>1</td></tr></table>		10	...	10	...	...	...	1	...	1	$Y=(T...T) Kol_A=10$ $Y=(T...T) Kol_B=10$ $Y=(T...T) Kol_C=10$ $k=3$ $S='A,B,C'$ (обр7)	Во всех массивах (A,B,C) все столбцы упорядочены; Максимальные размеры и количество упорядоченных элементов			
10	...	10																																				
...	...	...																																				
1	...	1																																				
10	...	10																																				
...	...	...																																				
1	...	1																																				
10	...	10																																				
...	...	...																																				
1	...	1																																				
6	$na=2, A (2 \times 2)$ <table><tr><td>10</td><td>-5</td></tr><tr><td>50</td><td>10</td></tr></table>		10	-5	50	10	$nb=2, B (2 \times 2)$ <table><tr><td>10</td><td>-5</td></tr><tr><td>50</td><td>10</td></tr></table>		10	-5	50	10	$nc=10, C (10 \times 10)$ <table><tr><td>10</td><td>...</td><td>10</td></tr><tr><td>...</td><td>...</td><td>...</td></tr><tr><td>1</td><td>...</td><td>1</td></tr></table>		10	...	10	...	...	...	1	...	1	$Y=(F,F) Kol_A=0$ $Y=(F,F) Kol_B=0$ $Y=(T...T) Kol_C=10$ $k=1$ $S='C'$ (обр7)	Только в одном массиве (C) есть упорядоченные столбцы													
10	-5																																					
50	10																																					
10	-5																																					
50	10																																					
10	...	10																																				
...	...	...																																				
1	...	1																																				
7	$na=2, A (2 \times 2)$ <table><tr><td>10</td><td>-5</td></tr><tr><td>50</td><td>10</td></tr></table>		10	-5	50	10	$nb=10, B (10 \times 10)$ <table><tr><td>10</td><td>...</td><td>10</td></tr><tr><td>...</td><td>...</td><td>...</td></tr><tr><td>1</td><td>...</td><td>1</td></tr></table>		10	...	10	...	...	...	1	...	1	$nc=2, C (2 \times 2)$ <table><tr><td>10</td><td>-5</td></tr><tr><td>50</td><td>10</td></tr></table>		10	-5	50	10	$Y=(F,F) Kol_A=0$ $Y=(T...T) Kol_B=10$ $Y=(F,F) Kol_C=0$ $k=1$ $S='B'$ (обр7)	Только в одном массиве (B) есть упорядоченные столбцы													
10	-5																																					
50	10																																					
10	...	10																																				
...	...	...																																				
1	...	1																																				
10	-5																																					
50	10																																					
8	$na=10, A (10 \times 10)$ <table><tr><td>10</td><td>...</td><td>10</td></tr><tr><td>...</td><td>...</td><td>...</td></tr><tr><td>1</td><td>...</td><td>1</td></tr></table>		10	...	10	...	...	...	1	...	1	$nb=2, B (2 \times 2)$ <table><tr><td>10</td><td>-5</td></tr><tr><td>50</td><td>10</td></tr></table>		10	-5	50	10	$nc=2, C (2 \times 2)$ <table><tr><td>10</td><td>-5</td></tr><tr><td>50</td><td>10</td></tr></table>		10	-5	50	10	$Y=(T...T) Kol_A=10$ $Y=(F,F) Kol_B=0$ $Y=(F,F) Kol_C=0$ $k=1$ $S='A'$ (обр7)	Только в одном массиве (A) есть упорядоченные столбцы													
10	...	10																																				
...	...	...																																				
1	...	1																																				
10	-5																																					
50	10																																					
10	-5																																					
50	10																																					
9	Далее тесты на аномальные ситуации (кроме аномалий при работе с файлами, если для них достаточно информации для проведения тестов в таблице аномальных ситуаций), если их рассматриваете.																																					
--	Аномальные ситуации при работе с файлами – см. пункт 7.Аномалии							Отсутствие необходимых параметров/файлов или недостаток прав, не числа в файле																														

## 9. Метод

Отделим ввод-вывод от обработки данных (собственно решения задачи).

*Указание:* не совмещать при решении *всех* задач этого семестра внутри одной процедуры ввод/вывод и обработку: либо только вывод, либо только поиск решения.

Выделим подзадачи:

1. Подзадача А0.1. Ввести исходные данные для одного указанного двумерного массива (размер и значения) из указанного (по имени) текстового файла. Если возникли аномальные ситуации при работе с исходным файлом, то вывести на консоль соответствующие сообщения, закрыть все файлы и присвоить переменной *InOk* значение *False*, иначе – *True*.
2. Подзадача А0.2. Вывести исходные данные по образцу 3 для одного указанного двумерного массива (размер и значения) в заранее открытый текстовый файл. Если возникли аномальные ситуации при работе с файлом, на консоль соответствующие сообщения, закрыть (если открыт) файл и присвоить переменной *OutOk* значение *False*, иначе – *True*.
3. Подзадача А0.3. Решение задачи Лаб.раб. №8: для одного указанного двумерного массива (размер и значения) создать массив *Y* и найти количество упорядоченных столбцов *kol*.
4. Подзадача А0.4. Вывод созданного массива *Y* и найденного количества в заранее открытый файл по образцу 4.

Подзадачи А0.1-4 реализуем в виде процедур и выполним поочередно для каждого из трех массивов А,В,С, очередное значение *kol* сохранять в соответствующем элементе массива *kol*.

Для решения подзадачи А0.3 в соответствии с условием подготовим **код в двух вариантах: с использованием а) процедуры общего вида; б) функции.**

5. Подзадача А0.5. На основе массива *kol* найти количество (*k*) массивов с упорядоченными столбцами и их имена (*S*).
6. Подзадача А0.6. Вывести окончательные результаты *k* и *S*, по образцу 5 или 6.

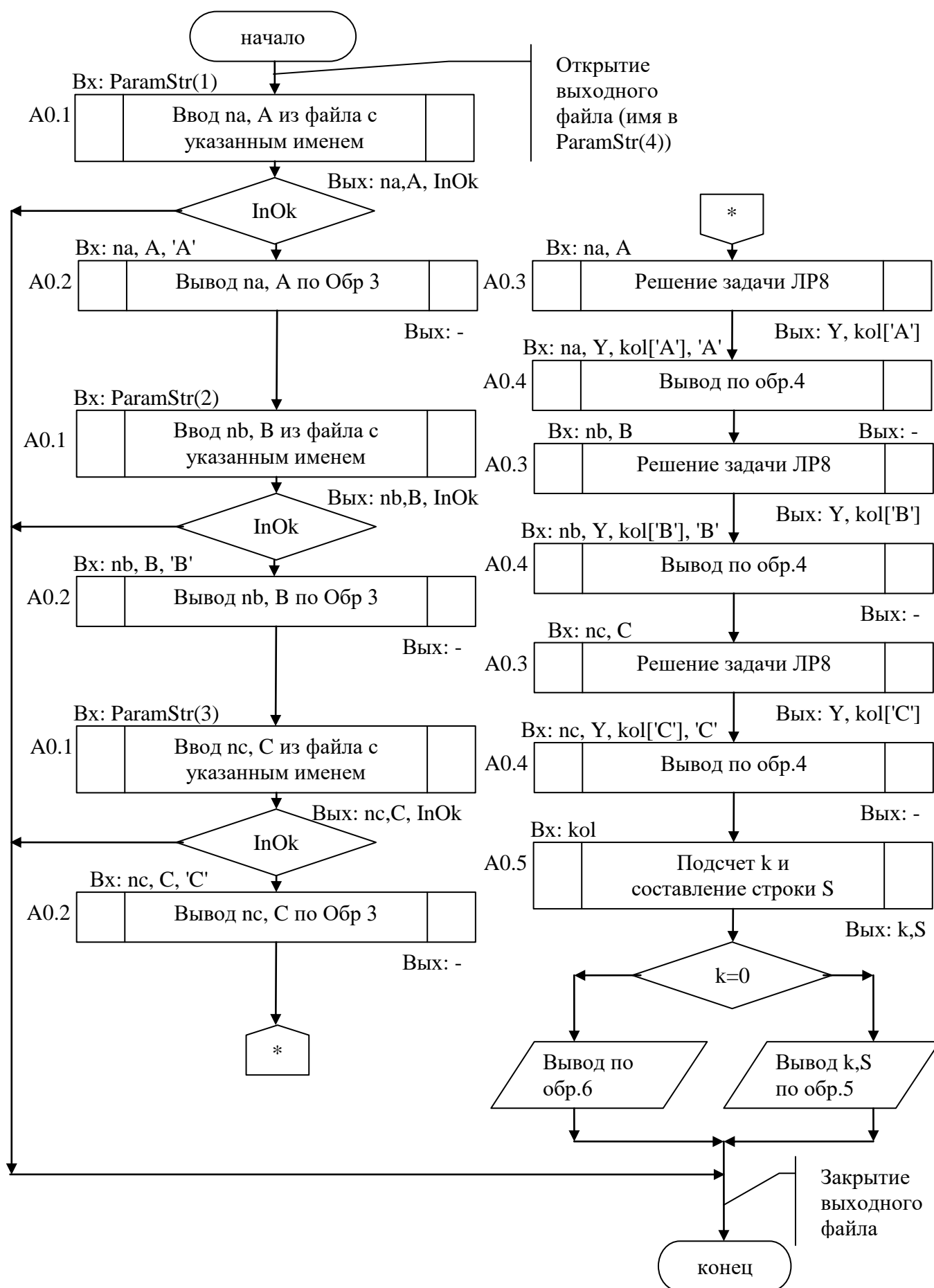
Подзадачу А0.5 реализуем в виде процедуры.

Открытие и закрытие каждого из трех файлов с исходными данными будем осуществлять внутри процедуры ввода (Подзадача А0.1).

Открытие файла для вывода исходных данных и результатов осуществим в самом начале программы, а закрытие – в самом конце после вывода всех результатов, за исключением закрытия в случае возникновения ошибок при вводе/выводе в подзадачах А0.1.

Подзадачу А0.6 реализуем сразу, так же как и открытие (с проверкой) выходного файла и весь вывод в него. А также проверку наличия не менее 4 параметров у программы.

## 10. Алгоритм (обработку аномалий можно не включать в алгоритм)



## 11. Программный код с процедурами-заглушками.

```

Program mas3;
{$APPTYPE CONSOLE}
Uses SysUtils, Windows;
Const nmax=10;
type matr = array[1..nmax, 1..nmax] of integer;
      mas = array[1..nmax] of boolean;
      kolmas = array ['A'..'C'] of byte;
      str5 = string[5];

procedure InputMatr(FName: String; out nx: byte; out X: matr; out InOk: Boolean);
var i,j:byte;
begin
    {Заглушка для A0.1. Откомментируйте нужный тест(пример кода см в Matrix.doc и Files-Exception.doc)
    {Тест 4} InOk:=true; nx:=2; X[1,1]:=10; X[1,2]:=-5; X[2,1]:=50; X[2,2]:=10;
    // {Аномалия} InOk:=false; writeln('Ошибка при вводе из файла ', FName);
end;

procedure OutputMatr(var F: TextFile; const nx: byte; var X: matr; const ch: char);
var i,j: byte;
begin
    writeln(F, 'Задан массив ', ch, ' (', nx, 'x', nx, '):'); {Вывод по обр3}
    for i:=1 to nx do
    begin
        for j:=1 to nx do write(F, X[i,j]:4, ' ');
        writeln(F);
    end;
end;

procedure OutputMas(var F: TextFile; const nx: byte; var Y: mas; const kol: byte; const ch: char);
var i:byte;
begin
    writeln(F, 'На основе массива ', ch, ' получен массив Y:'); {Вывод по обр4}
    for i:=1 to nx do write(F, Y[i]:5, ' '); writeln(F);
    writeln(F, 'kol', ch, ' = ', kol); {Вывод по обр4.2}
end;

```

```

(a) с процедурой
procedure lab8(const nx: byte; var X: matr; out Y:mas; out kol: byte);
var i,j: byte;
begin

```

```

    {Заглушка для A0.3. Откомментируйте нужный тест}
    {Тест 4} kol:=0; Y[1]:=false; Y[2]:=false;
    // {Тест 5} kol:=10; for i:=1 to 10 do Y[i]:=True;
end;

```

```

(б) с функцией
function lab8(const nx: byte; var X: matr; var Y:mas): byte;
var i,j, kol: byte;
begin

```

```

    {Заглушка для A0.3. Откомментируйте нужный тест}
    {Тест 4} kol:=0; Y[1]:=false; Y[2]:=false;
    // {Тест 5} kol:=10; for i:=1 to 10 do Y[i]:=True;
    lab8:=kol;
end;

```

Таблица соответствия параметров				
Формальные	Фактические			Типы
<i>nx</i>	<i>na</i>	<i>nb</i>	<i>nc</i>	<i>byte</i>
<i>X</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>matr</i>
<i>Y</i>	<i>Y</i>	<i>Y</i>	<i>Y</i>	<i>mas</i>
<i>kol</i>	<i>kol['A']</i>	<i>kol['B']</i>	<i>kol['C']</i>	<i>byte</i>

Таблица соответствия параметров				
Формальные	Фактические			Типы
<i>nx</i>	<i>na</i>	<i>nb</i>	<i>nc</i>	<i>byte</i>
<i>X</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>matr</i>
<i>Y</i>	<i>Y</i>	<i>Y</i>	<i>Y</i>	<i>mas</i>
<i>(lab8)</i>	<i>kol['A']</i>	<i>kol['B']</i>	<i>kol['C']</i>	<i>byte</i>

```

procedure FindKS(var kol: kolmas; out k: byte; out S: str5);
var ch: char;
begin {Заглушка для A0.5. Откомментируйте нужный тест или код}
    k:=0; S:='';
    {Тест 4} ;
    // {Тест 5} k:=3; S:='A,B,C';
    // {Поиск} for ch:='A' to 'C' do if kol[ch]>0 then
    //         if k=0 then begin k:=1; S:=ch; end
    //         else begin inc(k); S:=S+', '+ch; end;
end;

var

```



```

na,nb,nc, i,k: byte;
A,B,C: matr;
Y:mas;   kol:kolmas;
S: str5;
InOk, OutOk: Boolean;
dat, res: TextFile;
begin
  SetConsoleOutputCP(1251); writeln('Change this font to ''Lucida Console''');
  if ParamCount<4 then writeln('Мало параметров')
  else
    begin
      assignFile(res, ParamStr(4));
      OutOk:=True;
      Try // попытаемся создать файл
        rewrite(res);
      Except // если не удалось
        on E: Exception do
          Begin writeln('Невозможно создать файл ',ParamStr(4),': ', E.Classname,': ', E.Message);
            OutOk:=False;
          End;
        End;
      If OutOk then
        Begin
          writeln(res, ' ':15, 'Три массива. Вариант № 0'); { вывод по обр0}

          {ввод/вывод матриц по входной/выходной форме из трех текстовых файлов в один:}
          InputMatr(ParamStr(1), na, A, InOk);
          //   na:=InputMatr(ParamStr(1), A, InOk); { б) с функцией так }
          //   InOk:=InputMatr(ParamStr(1), na, A); { или эдак }
          if InOk then OutputMatr(res, na, A, 'A'); {вывод по обр3}

          if InOk then InputMatr(ParamStr(2), nb, B, InOk); {ввод по входной форме}
          if InOk then OutputMatr(res, nb, B, 'B'); {вывод по обр3}

          if InOk then InputMatr(ParamStr(3), nc, C, InOk); {ввод по входной форме}
          if InOk then OutputMatr(res, nc, C, 'C'); {вывод по обр3}
          End;

          if OutOk and InOk then
            begin
              lab8(na, A, Y, kol['A']); { а) с процедурой}
              //   kol['A']:= lab8(na, A, Y); { б) с функцией}
              OutputMas(res, na, Y, kol['A'], 'A');

              lab8(nb, B, Y, kol['B']); { а) с процедурой}
              //   kol['B']:= lab8(nb, B, Y); { б) с функцией}
              OutputMas(res, nb, Y, kol['B'], 'B');

              lab8(nc, C, Y, kol['C']); { а) с процедурой}
              //   kol['C']:= lab8(nc, C, Y); { б) с функцией}
              OutputMas(res, nc, Y, kol['C'], 'C');

              FindKS(kol, k, S); { а) с процедурой}
              //   k:=FindKS(kol, S); { б) с функцией}

              if k=0 then
                writeln(res, 'Не найдено ни одного массива с упорядоченными столбцами')
              else
                writeln(res, 'Всего найдено ', k, ' массив(а) с упорядоченными столбцами: ', S);
                CloseFile(res);
                Writeln('Ok. Файл ', ParamStr(4), ' готов. ');
              end;
            end;
          Writeln('Нажмите ENTER...':70);
          Readln {Задержка для просмотра сообщений об ошибках или 'Ok'}
        end.

```

## Задача №11 «Три массива». Абстракция A0.1

### 1. Постановка задачи

**Задание:** Дано имя файла *FName*, ввести от туда данные: *nx* – размер матрицы *X*, и значения элементов матрицы *X*, состоящей из *nx* строк и *nx* столбцов (см. форму ввода для абстракции A0). В случае, если файл не получится открыть или возникнут какие-либо ошибки при чтении из файла, то присвоить переменной *InOk* значение *FALSE* и выдать соответствующие сообщения (см. пункт 7.Аномалии для абстракции A0), иначе присвоить *InOk* значение *TRUE*.

Решение задачи выполнить в виде процедуры или в двух версиях: а) в виде процедуры общего вида; б) в виде функции.

### 5. Входная форма:

Обр1

```
<nx>
<X[ 1, 1]>...<X[ 1,nx]>
. . . . .
<X[nx, 1]>...<X[nx,nx]>
```

Обр2

## 11. Программный код с заглушкой:

```
procedure InputMatr(FName: String; out nx: byte; out X: matr; out InOk: Boolean);
var i,j:byte;
begin
    {Заглушка для A0.1. Откомментируйте нужный тест }
    {Тест 4} InOk:=true; nx:=2; X[1,1]:=10; X[1,2]:=-5; X[2,1]:=50; X[2,2]:=10;
    // {Аномалия} InOk:=false; writeln('Ошибка при вводе из файла ', FName);
end;
```

Таблица соответствия параметров				
Формальные	Фактические			Типы
<i>FName</i>	<i>ParamStr(1)</i>	<i>ParamStr(2)</i>	<i>ParamStr(3)</i>	<i>string</i>
<i>nx</i>	<i>na</i>	<i>nb</i>	<i>nc</i>	<i>byte</i>
<i>X</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>matr</i>
<i>InOk</i>	<i>InOk</i>	<i>InOk</i>	<i>InOk</i>	<i>boolean</i>

**Указание:** Программный код полностью написать самим, используя сведения, полученные на семинаре и лекции, про ввод матрицы (*Sem-5.pdf*) и информацию из лекции про обработку исключительных ситуаций при работе с файлами (*Files-Exceptions.pdf*).

## Задача №11 «Три массива». Абстракция A0.2

### 1. Постановка задачи

**Задание:** Дана файловая переменная *F*, связанная с заранее открытым для вывода текстовым файлом, а также: *nx* – размер матрицы *X*, и значения элементов матрицы *X*, состоящей из *nx* строк и *nx* столбцов, и имя матрицы – символ *ch*. Вывести матрицу в файл (см. форму вывода для абстракции A0, образец 3). Файл не закрывать.

Решение задачи выполнить в виде процедуры.

### 6. Выходная форма: (фрагмент, полностью см. в A0)

Обр3а

```
Задан массив <'A'> (<na>x<na>) :
<A[ 1, 1]>...<A[ 1,na]>
```

```
...
<A[na, 1]>...<A[na,na]>
```

```
Задан массив <'B'> (<nb>x<nb>) :
<B[ 1, 1]>...<B[ 1,nb]>
```

Обр3б

```
...
<B[nb, 1]>...<B[nb,nb]>
```

```
Задан массив <'C'> (<nc>x<nc>) :
<C[ 1, 1]>...<C[ 1,nc]>
```

Обр3с

```
...
<C[nc, 1]>...<C[nc,nc]>
```

## 11. Программный код:

```

procedure OutputMatr(var F: TextFile; const nx: byte; var X: matr; const ch: char);
var i,j: byte;
begin
  writeln(F, 'Задан массив ', ch, ' (', nx, 'x', nx, '); {вывод по обр3}
  for i:=1 to nx do
  begin
    for j:=1 to nx do write(F, X[i,j]:4, ' ');
    writeln(F);
  end;
end;

```

Таблица соответствия параметров				
Формальные	Фактические			Типы
<i>F</i>	<i>res</i>	<i>res</i>	<i>res</i>	<i>TextFile</i>
<i>nx</i>	<i>na</i>	<i>nb</i>	<i>nc</i>	<i>byte</i>
<i>x</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>matr</i>
<i>ch</i>	'A'	'B'	'C'	<i>char</i>

## Задача №11 «Три массива». Абстракция A0.3

### 1. Постановка задачи

**Задание:** Решение задачи из Лабораторной работы №8 для одного указанного двумерного массива (размер и значения) создать массив *Y* и найти количество упорядоченных столбцов *kol*

### 2. Уточненная постановка задачи.

Дан: размер матрицы (*nx*) и значения элементов матрицы *X*, состоящей из *nx* строк и *nx* столбцов. получить массив *Y* по правилу:  $Y_j$  получает значение *TRUE*, если элементы *j*-го столбца упорядочены по убыванию своих значений, иначе  $Y_j$  получает значение *FALSE*; и подсчитать также количество (*kol*) упорядоченных по убыванию столбцов в матрице.

Решение задачи выполнить в двух версиях: а) в виде процедуры общего вида; б) в виде функции.

### 11. Программный код:

С заглушкой:

```

{a) с процедурой}
procedure lab8(const nx: byte; var X: matr; out Y:mas; out kol: byte);
var i,j: byte;
begin
  {Заглушка для A0.3. Откомментируйте нужный тест}
  {Тест 4} kol:=0; Y[1]:=false; Y[2]:=false;
  // {Тест 5} kol:=10; for i:=1 to 10 do Y[i]:=True;
end;

```

Таблица соответствия параметров				
Формальные	Фактические			Типы
<i>nx</i>	<i>na</i>	<i>nb</i>	<i>nc</i>	<i>byte</i>
<i>X</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>matr</i>
<i>Y</i>	<i>Y</i>	<i>Y</i>	<i>Y</i>	<i>mas</i>
<i>kol</i>	<i>kol['A']</i>	<i>kol['B']</i>	<i>kol['C']</i>	<i>byte</i>

```

{б) с функцией}
function lab8(const nx: byte; var X: matr; var Y:mas): byte;
var i,j, kol: byte;
begin
  {Заглушка для A0.3. Откомментируйте нужный тест}
  {Тест 4} kol:=0; Y[1]:=false; Y[2]:=false;
  // {Тест 5} kol:=10; for i:=1 to 10 do Y[i]:=True;
  lab8:=kol;
end;

```

Таблица соответствия параметров				
Формальные	Фактические			Типы
<i>nx</i>	<i>na</i>	<i>nb</i>	<i>nc</i>	<i>byte</i>
<i>X</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>matr</i>
<i>Y</i>	<i>Y</i>	<i>Y</i>	<i>Y</i>	<i>mas</i>
<i>(lab8)</i>	<i>kol['A']</i>	<i>kol['B']</i>	<i>kol['C']</i>	<i>byte</i>

**Указание:** Программный код полностью написать самим, используя сведения, полученные на семинарах и лекциях про алгоритмы поиска количества и проверки условия (*Базовые-алгоритмы.pdf* и *Кодировании-алгоритмов.pdf*), а также про композицию алгоритмов в лекциях.

## Задача №11 «Три массива». Абстракция A0.4

### 1. Постановка задачи

**Задание:** Дана файловая переменная *F*, связанная с заранее открытым для вывода текстовым файлом, а также: *nx* – размер массива *Y*, и значения элементов массива *Y*, состоящего из *nx* элементов, и имя матрицы – символ *ch*, а также целое число *kol*. Вывести массив и число *kol* в текстовый файл (см. форму вывода для абстракции A0, образец 4). Файл не закрывать.

Решение задачи выполнить в виде процедуры.

## 6. Выходная форма: (фрагмент, полностью см. в A0)

Обр4.1a	На основе массива <'A'> получен массив Y: <Y[1]>...<Y[na]>
Обр4.2a	kol<'A'> = <kol['A']>
Обр4.1b	На основе массива <'B'> получен массив Y: <Y[1]>...<Y[nb]>
Обр4.2b	kol<'B'> = <kol['B']>
Обр4.1c	На основе массива <'C'> получен массив Y: <Y[1]>...<Y[nc]>
Обр4.2c	kol<'C'> = <kol['C']>

## 11. Программный код:

```
procedure OutputMas(var F: TextFile; const nx: byte; var Y: mas; const kol: byte; const ch: char);
var i: byte;
begin
  writeln(F, 'На основе массива ', ch, ' получен массив Y:'); {вывод по обр4}
  for i:=1 to nx do write(F, Y[i]:5, ' '); writeln(F);
  writeln(F, 'kol', ch, ' = ', kol); {вывод по обр4.2}
end;
```

Таблица соответствия параметров				
Формальные	Фактические			Типы
<i>F</i>	<i>Res</i>	<i>res</i>	<i>res</i>	<i>TextFile</i>
<i>nx</i>	<i>na</i>	<i>nb</i>	<i>nc</i>	<i>byte</i>
<i>Y</i>	<i>Y</i>	<i>Y</i>	<i>Y</i>	<i>mas</i>
<i>kol</i>	<i>kol['A']</i>	<i>kol['B']</i>	<i>kol['C']</i>	
<i>ch</i>	<i>'A'</i>	<i>'B'</i>	<i>'C'</i>	<i>char</i>

## Задача №11 «Три массива». Абстракция A0.5

### 1. Постановка задачи

**Задание:** Дан массив *kol* из трех чисел с индексами «A» «B» «C» - количество упорядоченных столбцов в соответствующих матрицах.

Вычислить:

*k* – количество матриц с упорядоченными столбцами и

*S* – строку с из именами матриц через запятую, например, «», «A», «C», «B,C», «A,B,C».

## 11. Программный код:

с заглушкой:

```
procedure FindKS(var kol: kolmas; out k: byte; out S: str5);
var ch: char;
begin {Заглушка для A0.5. Откомментируйте нужный тест}
  {Тест 4} k:=0; S:='';
  // {Тест 5} k:=3; S:='A,B,C';
end;
```

Программный код полностью:

```
procedure FindKS(var kol: kolmas; out k: byte; out S: str5);
var ch: char;
begin
  k:=0; S:='';
  for ch:='A' to 'C' do
    if kol[ch]>0 then
      if k=0 then
        begin // первый подходящий массив
          k:=1;
          S:=ch;
        end
      else
        begin // не первый
          inc(k);
        end
      end
  end;
```

```
        S:=S+', '+ch;  
    end;  
end;
```