



Министерство образования Республики Беларусь

Учреждение образования
«Гомельский государственный технический
университет имени П. О. Сухого»

Кафедра «Информационные технологии»

ОБРАБОТКА ОДНОМЕРНЫХ И ДВУХМЕРНЫХ МАССИВОВ НА VBA

ПОСОБИЕ

**по выполнению контрольных и лабораторных
работ по дисциплинам «Информатика» и «Основы
информатики и вычислительной техники»
для студентов экономических специальностей
дневной и заочной форм обучения**

Электронный аналог печатного издания

Гомель 2007

УДК 004.43(075.8)
ББК 32.793-018.1я73
О-23

*Рекомендовано к изданию научно-методическим советом
факультета автоматизированных и информационных систем
ГГТУ им. П. О. Сухого
(протокол № 9 от 10.05.2006 г.)*

Авторы-составители: *Н. В. Водополова, С. А. Чабуркина*

Рецензент: начальник сектора автоматизированных систем управления вычислительного центра ГГТУ им. П. О. Сухого *Н. С. Шестакова*

Обработка одномерных и двумерных массивов на VBA : пособие по выполнению контрол. и лаб. работ по дисциплинам «Информатика» и «Основы информатики и вычислительной техники» для студентов экон. специальностей днев. и заоч. форм обучения / авт.-сост.: Н. В. Водополова, С. А. Чабуркина. – Гомель : ГГТУ им. П. О. Сухого, 2007. – 33 с. – Систем. требования: PC не ниже Intel Celeron 300 МГц ; 32 Mb RAM ; свободное место на HDD 16 Mb ; Windows 98 и выше ; Adobe Acrobat Reader. – Режим доступа: <http://gstu.local/lib>. – Загл. с титул. экрана.

ISBN 978-985-420-635-6.

Рассмотрены вопросы обработки одномерных и двумерных массивов на языке VBA. Представлены алгоритмы ввода и вывода элементов массива, типовые алгоритмы обработки массивов и их программная реализация на языке программирования.

Для студентов экономических специальностей дневной и заочной форм обучения.

УДК 004.43(075.8)
ББК 32.793-018.1я73

ISBN 978-985-420-635-6

© Водополова Н. В., Чабуркина С. А.,
составление, 2007
© Учреждение образования «Гомельский
государственный технический университет
имени П. О. Сухого», 2007

1. ОРГАНИЗАЦИЯ РАБОТЫ С ОДНОМЕРНЫМИ МАССИВАМИ

1.1. Объявление одномерного массива

Массив – это структурированный тип данных, состоящий из фиксированного числа элементов, имеющих один и тот же тип.

Например:

- $A = (3, 4, 18, 6, 0, 5)$ – массив целых чисел;
- $Name = ("Анна", "Людмила", "Елена", "Галина")$ – массив строковых значений;
- $W = (1.5, 2.3, -1.9)$ – массив вещественных чисел.

Прежде чем начать работать с элементами массива, массив необходимо объявить – неявное объявление массива недопустимо.

Синтаксис объявления одномерного массива:

$Dim <Имя массива> (<H_гр_индекса> To <B_гр_индекса >) As <Тип>$

Все элементы синтаксиса обязательны:

Имя массива – имя массива, удовлетворяющее стандартным правилам именования переменных;

H_гр_индекса; *B_гр_индекса* – нижнее (минимальное) и верхнее (максимальное) значение индекса;

Тип – тип данных массива.

Например, инструкции

- $Dim Plan (1 To 17) As Integer$
- $Dim F (100 To 170) As Single$

объявляют:

– переменную *Plan* типа массив целых чисел, значение индекса массива изменяется от 1 до 17 включительно;

– переменную *F* типа массив вещественных чисел, значение индекса массива изменяется от 100 до 170 включительно.

При объявлении переменной типа «массив» происходит резервирование памяти в соответствии:

- с типом элементов массива;
- с максимально возможным количеством элементов массива.

Доступ к отдельному элементу массива осуществляется путем индексирования элементов массива.

Индекс элемента массива – это его порядковый номер в массиве.

Для обращения к конкретному элементу массива необходимо указать имя массива и в круглых скобках индекс элемента.

Алгебраическая запись	Запись на VBA
a_1	$A(1)$
a_3	$A(3)$
a_i	$A(i)$

Например:

➤ $A(3) = 56$ – третьему элементу массива A присвоено значение «56»;

➤ $Name(0) = "Anna"$ – нулевому элементу массива $Name$ присвоено значение «Anna»;

➤ $B(i) = Input("B(" & i & ")=")$ – i -му элементу массива B присваивается значение, которое будет введено с клавиатуры. Очевидно, что значение переменной i должно быть определено до оператора присваивания.

Индексы могут представлять собой выражения любого скалярного типа, кроме вещественного. Возможно, например, обращение $A(I + 1)$ или $A(N - i + 1)$ для целочисленных переменных i и N соответственно.

Тип индекса определяет границы изменения значений индекса. Для типа *Byte* – максимальное значение индекса – 255, для типа *Integer* – 32767.

1.2. Ввод элементов массива

Процедура ввода значений элементов массива разбивается на два этапа:

1. Ввод размерности массива, т. е. того количества элементов массива, с которым пользователь хочет работать. Это значение не должно превышать значение $V_gr_индекса$, указанного при его объявлении.

2. Поочередный ввод значений элементов массива. Если игнорировать этот этап, в зарезервированных при объявлении массива ячейках памяти будут находиться нулевые значения (для числовых типов), а не те, с которыми хотел бы иметь дело пользователь.

Например, для массива x , объявленного как $Dim x(1 To 10) As Byte$, возможен ввод не более 10 целых чисел, значение которых не превышает 255. То есть вводимое значение размерности массива N должно быть не больше 10 и $x(i)$ – любое неотрица-

тельное число до 255 включительно, т. к. тип элементов массива объявлен *Byte*. Если при вводе данных ввести некорректные значения, программа завершит свою работу аварийно.

Процесс ввода элементов массива графически представляется следующим образом (рис. 1.1):

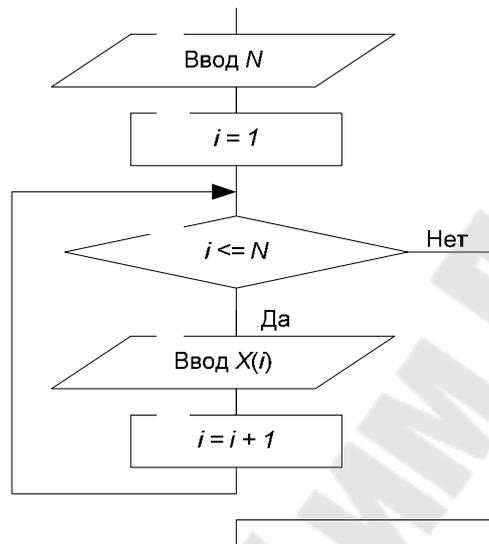
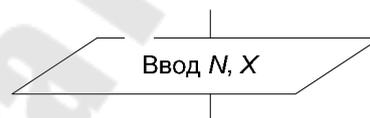


Рис. 1.1. Схема алгоритма ввода элементов массива

Поскольку это довольно очевидное и распространенное действие, то обычно этот фрагмент графической схемы алгоритма оформляется в виде одного блока:



где N – размерность массива, а X – имя массива, элементы которого вводятся.

В программе эти действия реализуются следующим образом:

```

Sub Pr()
  Dim N As Byte
  Dim X(1To25) As Integer
  N = InputBox("Введи размерность массива")
  For I = 1 To N
    X(i) = InputBox("Введи X("& i &"-й элемент массива")
  Next
End Sub
  
```

При выполнении этого фрагмента программы происходит следующее:

1. В памяти компьютера по инструкции

Dim X(1To25) As Integer

находится свободная область объемом 50 байт, т. к. каждое целое число занимает 2 байта, а максимальное количество элементов в объявляемом массиве X – 25. Этой области памяти присваивается имя X , а вся область структурируется на 25 компонентов по 2 байта каждый. Всем элементам массива присваивается нулевое значение.

2. По инструкции

N = InputBox("Введи размерность массива")

определяется конкретное количество компонент из области памяти X , с которым хочет работать пользователь программы.

3. С помощью цикла

For i = 1 To N

X(i) = InputBox ("Введи x("& i &") – й элемент массива")

Next

осуществляется поочередное наполнение N компонент области памяти X конкретными значениями.

1.3. Вывод элементов массива

Графически вывод значений элементов массива может быть представлен в виде следующего фрагмента схемы (рис. 1.2):

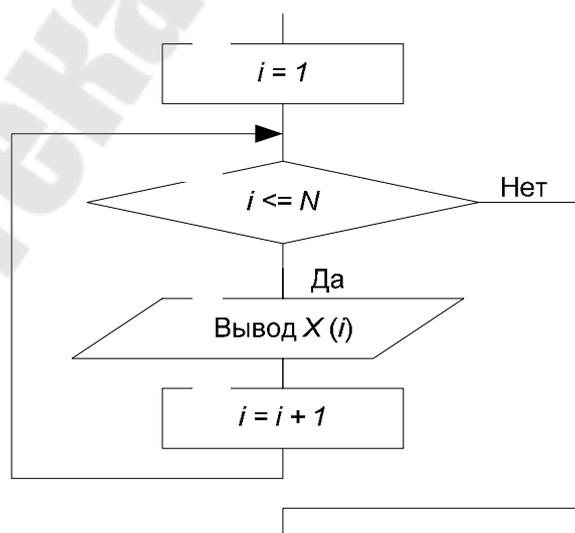


Рис. 1.2. Схема алгоритма вывода значений элементов массива

Поскольку, так же как и ввод элементов массива, их вывод – процесс очевидный и распространенный, то в графических схемах алгоритмов он обычно изображается одним блоком:



Программная реализация вывода элементов массива зависит от метода организации вывода. Мы будем осуществлять вывод элементов массива только на рабочий лист Excel. Вывод элементов массива может осуществляться в разных строках одного столбца рабочего листа Excel (построчный вывод) или в разных столбцах одной строки.

Построчный вывод элементов массива:

```
Cells (1, 1) = "Массив:"
For i = 1 To N
    Cells (1+i, 1) = X(i)
Next
```

Вывод элементов массива в одной строке:

```
Cells (1, 1) = "Массив:"
For i = 1 To N
    Cells (2, i) = X(i)
Next
```

Реализация построчного вывода элементов массива и вывода элементов массива в одной строке представлена на рис. 1.3.

	A	B	C
1	Исходный массив:		
2		5	
3		9	
4		0	
5		4	
6		2	
7		3	
8			

	A	B	C	D	E	F	G	H
1	Исходный массив:							
2		5	8	9	1	4	6	
3								
4								

Рис. 1.3. Вывод элементов массива

1.4. Типовые алгоритмы обработки одномерных массивов

Типовыми алгоритмами обработки одномерных массивов являются: определение суммы, произведения, количества, среднего арифметического элементов массивов; поиск максимальных и минимальных элементов и другой информации в массиве; формирование нового массива.

1.4.1. Определение количества, суммы, среднего арифметического элементов массива

Определение количества k и суммы значений S элементов массива, удовлетворяющих некоторому условию, – это процесс накопления в переменной k единиц, а в переменной S значений искомым элементов массива по формулам $k = k + 1$ и $S = S + A(i)$ соответственно.

Очевидно, что начальное значение переменных k и S должны быть равны нулю, иначе будет получен неверный результат.

Для вычисления среднего значения элементов массива необходимы две величины: суммы (S) и количества элементов (K). Среднее вычисляется по формуле

$$Sr = \frac{S}{K}.$$

Поскольку деление на нуль вызывает аварийное завершение работы программы, то перед вычислением среднего значения проверка значения K на нуль должна быть выполнена обязательно.

Пример 1.1. Найти среднее значение чисел одномерного массива, стоящих на четных местах и принадлежащих промежутку $(x; y]$.

Графическая схема алгоритма решения задачи представлена на рис. 1.4, результаты тестирования программы – на рис. 1.5.

```
Sub Pr_1_1()
  Dim N As Byte, A(1 To 10) As Integer
  Dim i As Byte, S As Integer, K As Byte
  Dim x As Integer, y As Integer
  Dim Sr As Single
  'Ввод и вывод исходных данных
  N = InputBox("Введи размерность массива")
  For i = 1 To N
    A(i) = InputBox("Введи значение A(" & i & ")-го элемента")
  Next
  x = InputBox("Введи значение нижней границы промежутка")
  y = InputBox("Введи значение верхней границы промежутка")
  Cells(1, 1) = "Исходный массив:"
  For i = 1 To N
    Cells(2, 1 + i) = A(i)
  Next
  'Определение среднего значения
  S = 0: K = 0
  For i = 2 To N Step 2
    If A(i) > x And A(i) <= y Then S = S + A(i): K = K + 1
```

```

Next
'Вывод результата
If K = 0 Then
    Cells(3, 1) = "На четных местах нет чисел из промежутка ("
                & x & ";" & y & "]"
Else
    Sr = S / K
    Cells(3, 1) = "Среднее значение чисел, стоящих на четных местах"
    Cells(4, 1) = " и принадлежащих промежутку (" & x & ";" & y & "]"
    Cells(5, 1) = "равно " & Sr
End If
End Sub

```

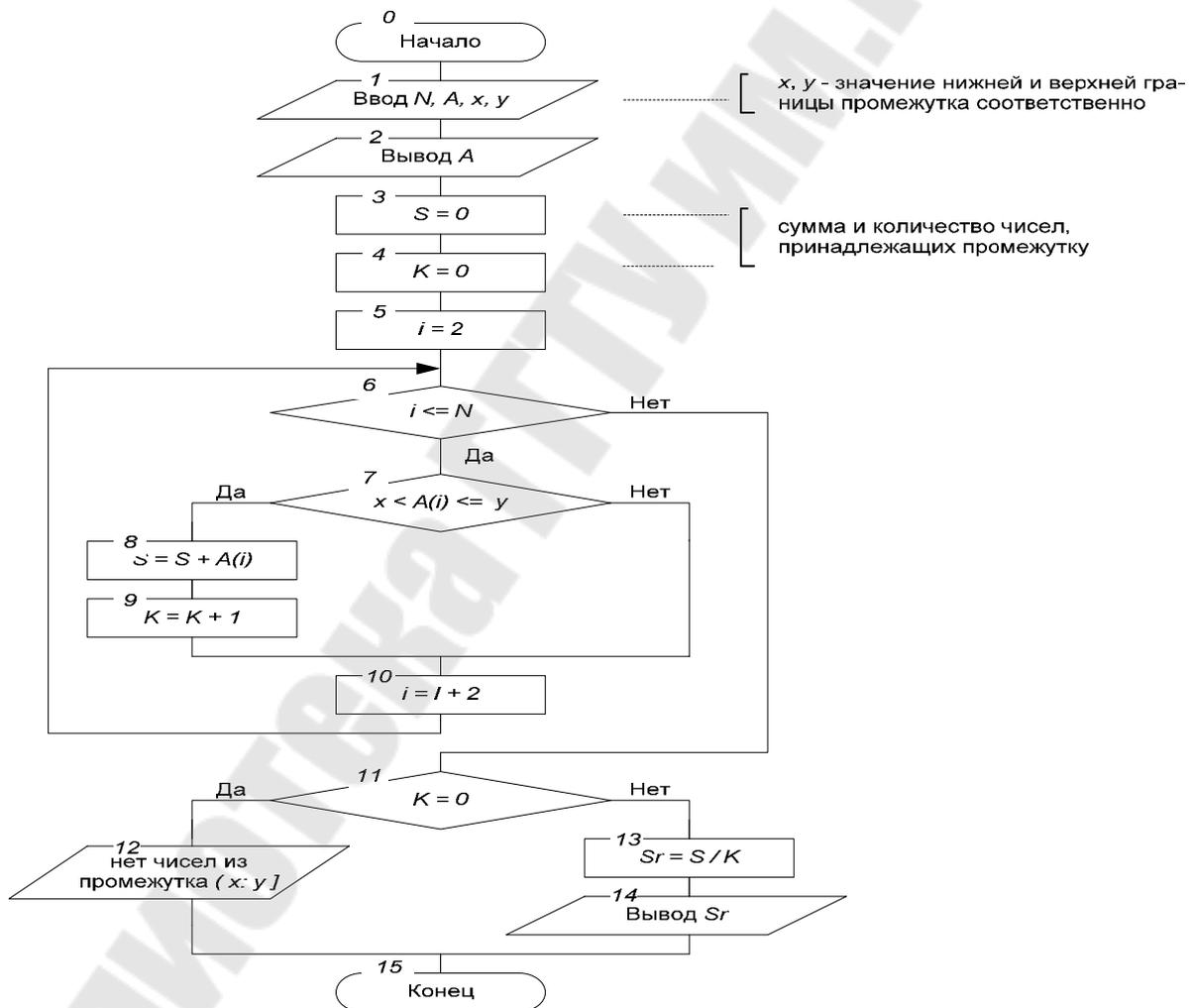


Рис 1.4. Схема алгоритма примера 1.1

А	В	С	Д	Е	Р	В	И	Т
1	Исходный массив							
2		4	5	6	7	8	9	0
3	Произведение элементов массива на промежутке [3;7]							
4								

А	В	С	Д	Е	Р	В	И	Т
1	Исходный массив							
2		0	3	3	0	0	3	
3	Среднее значение чисел, стоящих на четных местах в представляющих промежутке [1;6]							
4	равно 3							
5								

А	В	С	Д	Е	Р	В	И	Т
1	Исходный массив							
2		1	1	2	2	1	2	
3	Произведение элементов массива, стоящих на четных местах в представляющих промежутке [1;5]							
4	равно 4							
5								

Рис. 1.5. Результаты работы программы Pr_1_1

1.4.2. Определение произведения элементов массива

Произведение элементов массива накапливается в соответствии со следующей формулой:

$$p = p * A(i).$$

Очевидно, что начальное значение переменной p должно быть равно 1. Так же, как и при вычислении значения суммы элементов, при определении произведения элементов массива может возникнуть ситуация, когда результат вычислений, т. е. последнее значение переменной p , и ее начальное значение могут совпадать. Например, для массива $A = (1, 1, 0, -1, -3)$ произведение положительных чисел равно 1. Поэтому при вычислении произведения элементов массива необходимо использование дополнительной переменной, с помощью которой можно будет сделать вывод о наличии в массиве элементов, удовлетворяющих условию задачи.

Пример 1.2. Найти произведение чисел одномерного массива, не принадлежащих промежутку $[x; y]$ и стоящих на местах, кратных 5.

Алгоритм представлен на рис. 1.6, результаты тестирования программы – на рис. 1.7.

В качестве дополнительной переменной используется переменная W логического типа, которая может принимать два значения: *True* – Истина и *False* – Ложь. Значение *True* – начальное, *False* – если в массиве встретилось хотя бы одно число, не принадлежащее промежутку $[x; y]$.

Sub Pr_1_2()

Dim N As Byte, A(1 To 10) As Integer

Dim i As Byte, P As Integer

Dim x As Integer, y As Integer

```

Dim W As Boolean
'Ввод и вывод исходных данных
N = InputBox("Введи размерность массива")
For i = 1 To N
    A(i) = InputBox("Введи значение A(" & i & ") -го элемента")
Next
x = InputBox("Введи значение нижней границы промежутка")
y = InputBox("Введи значение верхней границы промежутка")
'Обработка данных
If N < 5 Then
    Cells(3, 1) = "В массиве меньше 5 чисел"
Else
    'Нахождение произведения
    W = True
    P = 1
    For i = 5 To N Step 5
        If A(i) < x Or A(i) > y Then P = P * A(i): W = False
    Next
'Вывод результата
If W = True Then
    Cells(3, 1) = "На местах, кратных 5, все числа из промежутка [" & _
        & x & ";" & y & "]"
Else
    Cells(3, 1) = "Произведение чисел, не принадлежащих"
    Cells(4, 1) = "промежутку [" & x & ";" & y & "], равно " & P
End If
End If
Cells(1, 1) = "Исходный массив:"
For i = 1 To N
    Cells(2, 1 + i) = A(i)
Next
End Sub

```

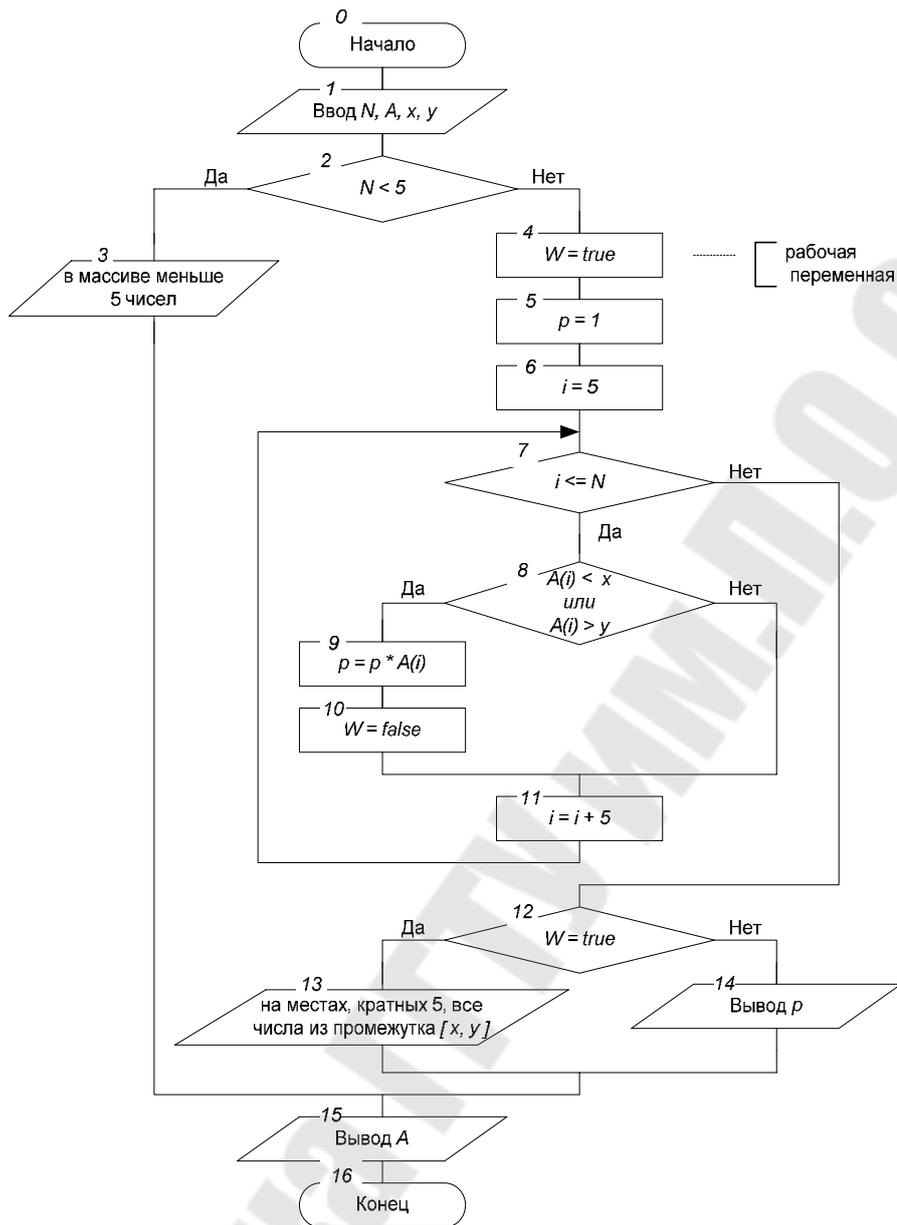


Рис. 1.6. Схема алгоритма примера 1.2

	A	B	C	D	E	F	G	H	I	
1	Исходный массив:									
2	2	3	3	2	2	4	2	3	3	
3	На местах, кратных 5, все числа из промежутка [4,9]									
4										
5										

	A	B	C	D	E	F	G	H	I	J	K	L
1	Исходный массив:											
2	2	2	2	2	2	4	2	2	2	2	6	
3	Произведение чисел, не принадлежащих промежутку [4,6], равно 16											
4												
6												

	A	B	C	D	E	F	G	H	I	J	K	L
1	Исходный массив:											
2	2	2	2	2	2	2	8	8	8	8	8	
3	Произведение чисел, не принадлежащих промежутку [4,6], равно 16											
4												
6												

Рис. 1.7. Результаты работы программы Pr_1_2

1.4.3. Поиск экстремумов

Экстремум – это минимальное и максимальное значение в группе однородных элементов.

Нахождение экстремумов в одномерном массиве сводится к выполнению следующих действий:

1. В качестве значения экстремума запоминается значение первого элемента массива.

2. Начиная со следующего, второго, все элементы массива поочередно сравниваются с текущим значением экстремума. Если очередной элемент массива превышает (при нахождении максимума) или является меньшим (при нахождении минимума), значение экстремума заменяется значением обрабатываемого элемента массива.

На рис. 1.8 изображен фрагмент графической схемы, отображающий алгоритм нахождения в одномерном массиве максимума. Если в условии ветвления операцию отношения «больше» заменить операцией отношения «меньше», получим алгоритм нахождения минимума.

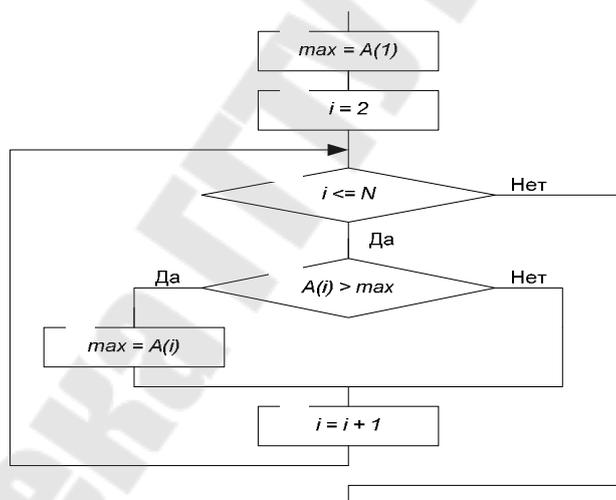


Рис. 1.8. Фрагмент схемы алгоритма нахождения максимума

Для определения местоположения экстремума в массиве необходимо использовать дополнительную переменную.

Пример 1.3. В одномерном массиве значение минимального элемента заменить числом -1000 .

Графическая схема алгоритма изображена на рис. 1.9, результаты тестирования программы – на рис. 1.10.

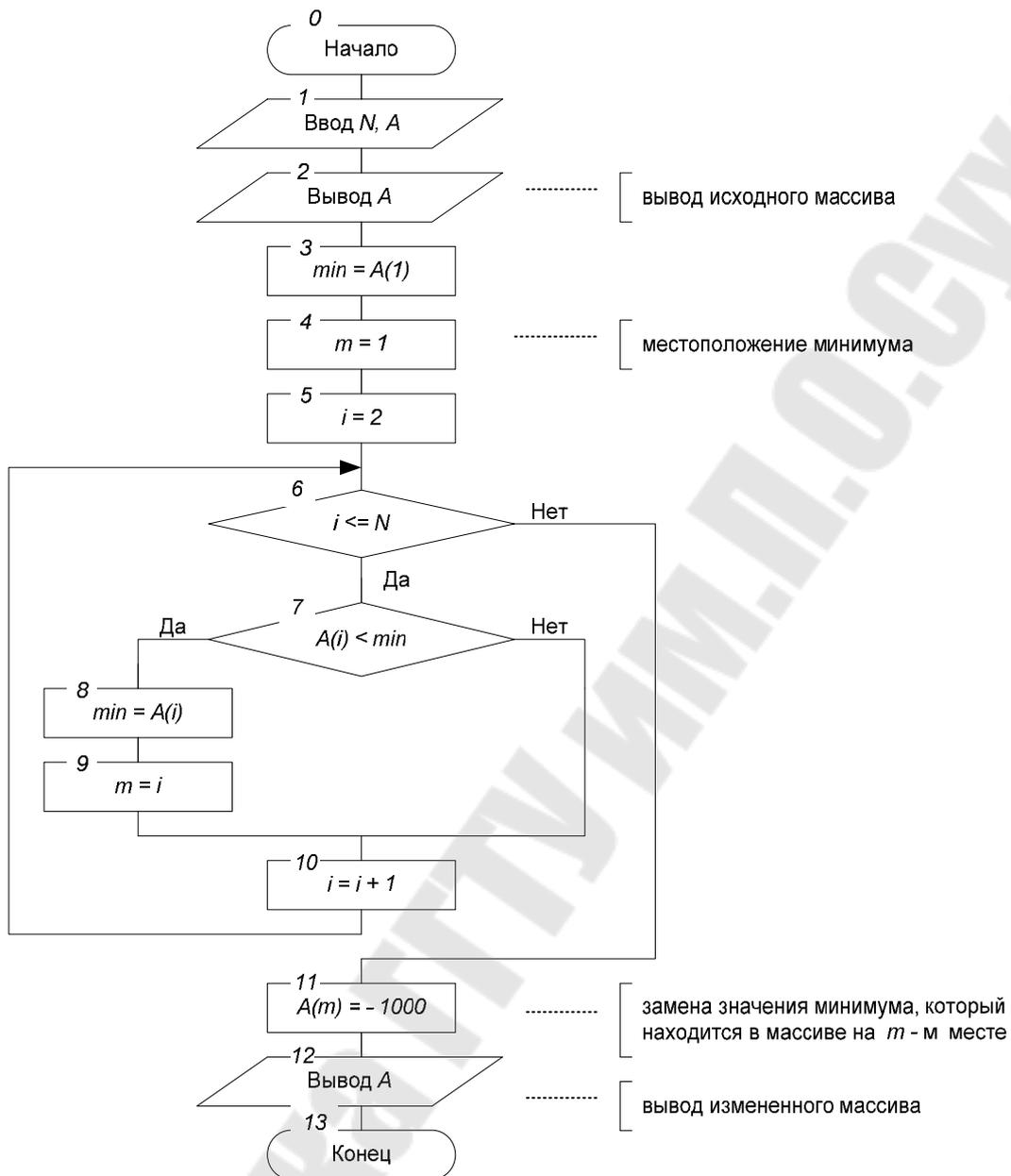


Рис. 1.9. Схема алгоритма примера 1.3

Sub Pr_1_3()

Dim N As Byte, A(1 To 10) As Integer

Dim min As Integer, m As Byte, i As Byte

'Ввод и вывод исходных данных

N = InputBox("Введи размерность массива", "Пример 1.3")

For i = 1 To N

A(i) = InputBox("Введи значение " & i & "-го элемента массива")

Next

Cells(1, 1) = "Исходный массив:"

For i = 1 To N

```

Cells(2, 1 + i) = A(i)
Next
'Нахождение минимума и его местоположения
min = A(1): m = 1
For i = 2 To N
    If A(i) < min Then min = A(i): m = i
Next
'Замена значения минимума и вывод измененного массива
A(m) = -1000
Cells(4, 1) = "Измененный массив"
For i = 1 To N
    Cells(5, 1 + i) = A(i)
Next
End Sub

```

Рис. 1.10. Результаты работы программы Pr_1_3

В массиве может быть несколько минимальных и максимальных значений. Например, в массиве $A=(0, -1, 8, -1, 8, -1)$ у трех элементов значения соответствуют минимальному (-1) и у двух – максимальному (8) значению. Поэтому при нахождении экстремумов в таких случаях говорят о левом (первом) или правом (последнем) минимуме или максимуме.

Для нахождения левых экстремумов массива используются операции отношения «больше» или «меньше». Если в условии задачи специально не оговорено, какой именно требуется найти экстремум, то имеется в виду левый. Алгоритм рис. 1.9 является именно таким. В четвертом и пятом тестах (рис. 1.10) найдено местоположение именно левого минимального элемента.

Для нахождения правых экстремумов массива необходимо использовать операции отношения «больше или равно» или «меньше или равно».

1.4.4. Формирование нового массива

Процесс формирования нового массива из данных исходного массива, удовлетворяющих конкретному условию, связан с просмотром исходного массива и присваиванием значения его очередного элемента, удовлетворяющего некоторому критерию, элементу формируемого массива.

Например, если из положительных элементов массива

$$A = (0, -1, 4, -5, 0, 7, 8, 3, 0)$$

необходимо сформировать новый массив, то, очевидно, что его содержимое будет таким:

$$B = (4, 7, 8, 3).$$

Случай, когда все элементы исходного массива удовлетворяют условию отбора, скорее является исключением, чем правилом. Поскольку не все элементы исходного массива могут соответствовать условию отбора, то:

- 1) размерность формируемого массива не больше размерности исходного массива. Для приведенного выше примера: размерность исходного массива (массив A) – 9, сформированного (массив B) – 4;
- 2) местоположение конкретного элемента в исходном и сформированном массиве различно.

Например, число 4 в исходном массиве является третьим элементом, а в сформированном – первым. Поэтому для индексирования элементов формируемого массива должна использоваться самостоятельная переменная, а не та, которая используется при просмотре элементов исходного массива;

3) переменной индексирования элементов формируемого массива необходимо присвоить первоначальное значение 0. Изменение индекса осуществляется до присваивания значения элементу формируемого массива;

4) значение переменной индексирования элементов формируемого массива может быть использовано:

– при принятии решения: массив был сформирован или нет. Если ее первоначальное значение не изменилось, то массив сформирован не был;

– для определения размерности сформированного массива.

Алгоритм рис. 1.11 реализован в программе *Prim_f*, результаты тестирования которой представлены на рис. 1.12.

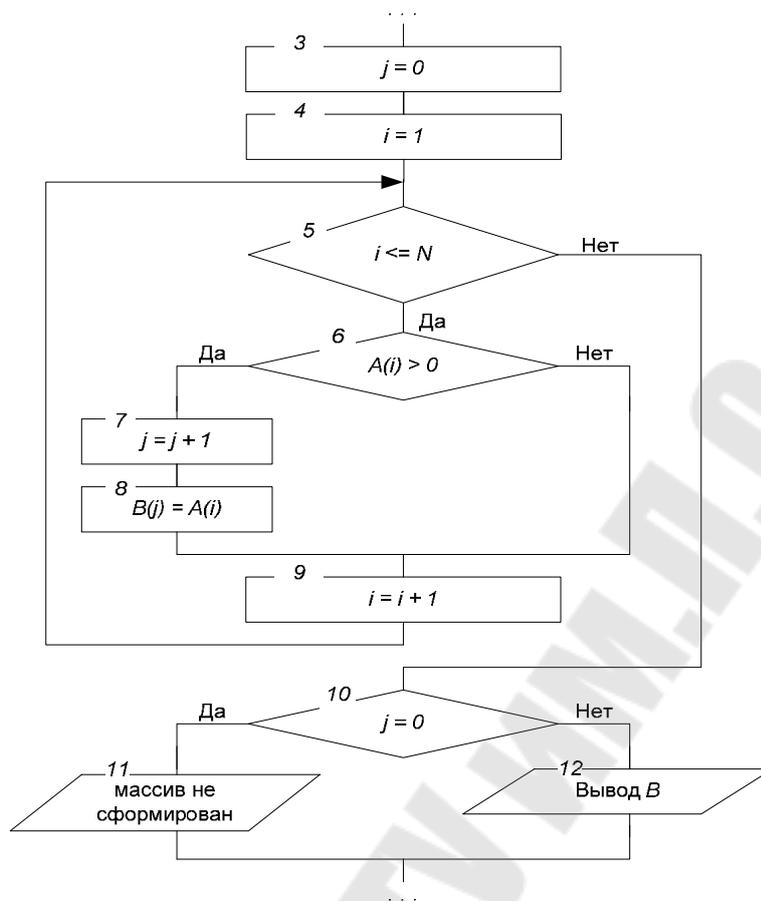


Рис. 1.11. Схема алгоритма формирования нового массива

Sub Prim_f()

Dim N As Byte, A(1 To 10) As Integer, B(1 To 10) As Integer

Dim j As Byte, i As Byte

'Ввод и вывод исходных данных

N = InputBox("Введи размерность массива")

For i = 1 To N

A(i) = InputBox("Введи значение " & i & "-го элемента массива")

Next

Cells(1, 1) = "Исходный массив:"

For i = 1 To N

Cells(2, 1 + i) = A(i)

Next

'Формирование массива

j = 0

For i = 1 To N

If A(i) > 0 Then

j = j + 1

```

    B(j) = A(i)
  End If
Next
'Вывод сформированного массива
If j = 0 Then
  Cells(4, 1) = "Массив не сформирован, т.к. в исходном массиве " _
    & "нет положительных чисел"
Else
  Cells(4, 1) = "Сформированный массив:"
  For i = 1 To j
    Cells(5, 1 + i) = B(i)
  Next
End If
End Sub

```

	A	B	C	D	E	F	G	H
1	Исходный массив:							
2		0	0	-5	-7	0		
3								
4	Массив не сформирован, т.к. в исходном массиве нет положительных чисел							
5								
6								
7								

	A	B	C	D	E	F	G	H	I	J	K
1	Исходный массив:										
2		0	-1	4	-5	0	7	0	3	0	
3											
4	Сформированный массив:										
5		4	7	8	3						
6											

Рис. 1.12. Результаты программы формирования массива

1.4.5. Поиск информации в массиве

Очень часто необходимо не столько определить, например, количество отрицательных чисел, а выявить просто факт наличия их в обрабатываемом массиве. При решении подобного рода задач совсем необязательно просматривать все элементы обрабатываемого массива до конца. Как только будет определен факт наличия конкретной ситуации, следует закончить обработку массива. Для этого необходимо:

1. Присвоить дополнительной переменной любое значение. Если переменная целочисленная – это может быть, например, нуль, если логическое – *true* (блоки 4 графической схемы алгоритма рис. 1.13, *a* и *б* соответственно).

2. Для входа в цикл использовать сложное логическое выражение (блоки 6), состоящее из двух простых условий, объединенных операцией «логическое И» (оператор *and*). Это обеспечит вход в цикл только в том случае, если оба условия будут истинны.

3. Изменить значение дополнительной переменной при обнаружении конкретной ситуации. Для алгоритма рис. 1.13 – это наличие отрицательного числа (блока 8).

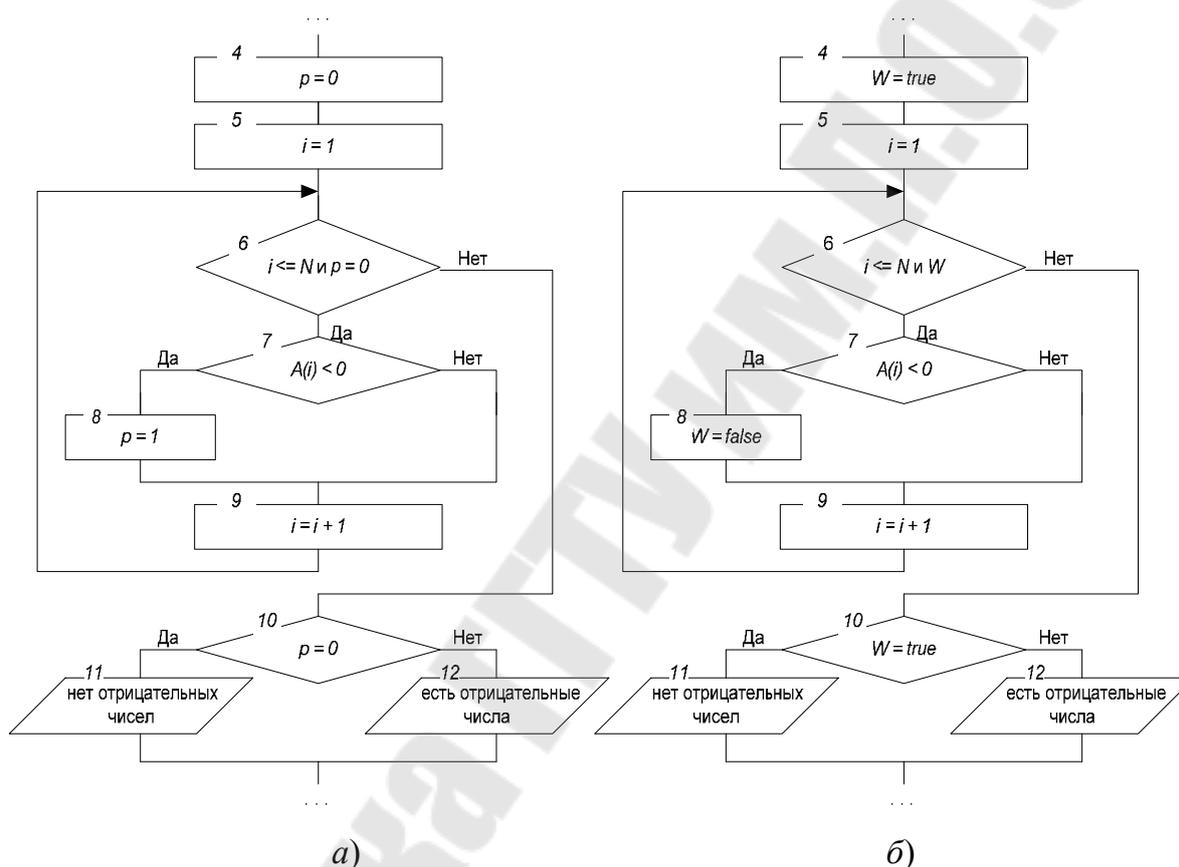


Рис. 1.13. Схемы алгоритмов поиска информации в массиве

4. Проанализировать значение дополнительной переменной. Если ее значение не изменилось, то это означает, например, для рассматриваемого алгоритма, что в массиве нет отрицательных чисел.

Алгоритм *a* рис. 1.13 реализован в программе *Pr_poisk*, результаты ее тестирования – рис. 1.14.

Sub Pr_poisk()

Dim N As Byte, A(1 To 10) As Integer

Dim i As Byte, P As Byte

'Ввод и вывод исходных данных

```

N = InputBox("Введи размерность массива")
For i = 1 To N
    A(i) = InputBox("Введи значение A(" & i & ")")
Next
Cells(1, 1) = "Исходный массив:"
For i = 1 To N
    Cells(2, 1 + i) = A(i)
Next
'Нахождение первого отрицательного числа
P = 0
i = 1
While i <= N And P = 0
    If A(i) < 0 Then P = 1
    i = i + 1
Wend
'Вывод результатов
If P = 0 Then
    Cells(4, 1) = "В массиве нет отрицательных чисел"
Else
    Cells(4, 1) = "В массиве есть хотя бы одно отрицательное число"
End If
End Sub

```



Рис. 1.14. Результаты поиска информации в массиве

2. ОРГАНИЗАЦИЯ РАБОТЫ С ДВУХМЕРНЫМИ МАССИВАМИ

2.1. Объявление двумерного массива

Двухмерный массив или матрица – это упорядоченная последовательность величин одного типа, имеющих одно имя, но различающихся индексами. Каждая величина в последовательности называется элементом массива. Все элементы матрицы в памяти компьютера расположены последовательно один за другим. Однако матрицу можно представить в виде двумерной таблицы, в каждой ячейке которой находится элемент массива. Положение элемента в массиве определяют два целых числа (индекса) – номер строки и номер столбца. Размерность матрицы определяется двумя целыми числами – числом строк и числом столбцов.

	5	-7	0	127
	12	-4	77	8
	87	-1	5	56

Рис. 2.1. Представление матрицы в виде двумерной таблицы

Представленный на рис. 2.1 массив состоит из 3 строк и 4 столбцов. Первый элемент массива $A_{11} = 5$, последний элемент $A_{34} = 56$, элемент $A_{23} = 77$.

Синтаксис объявления двумерного массива:

$Dim \langle \text{Имя массива} \rangle (\langle N_гр_индекса1 \rangle To \langle B_гр_индекса1 \rangle, \langle N_гр_индекса2 \rangle To \langle B_гр_индекса2 \rangle) As \langle \text{Тип} \rangle$

Все элементы синтаксиса обязательны:

– *Имя массива* – имя массива, удовлетворяющее стандартным правилам именования переменных;

– $N_гр_индекса1$, $B_гр_индекса1$ определяют диапазон изменения первого индекса (номера строки); $N_гр_индекса2$, $B_гр_индекса2$ – диапазон изменения второго индекса (номера столбца).

Если $N_гр_индекса1 = N_гр_индекса2 = 1$, то значения $B_гр_индекса1$, $B_гр_индекса2$ определяют максимальное число строк и столбцов матрицы;

– *Тип* – тип элементов массива.

Например, инструкции

➤ *Dim X (1 To 10, 1 To 10) As Integer*

➤ *Dim A (1 To 5, 1 To 3) As Single*

объявляют:

– двумерный массив целых чисел X , значения каждого индекса (номера строки и номера столбца) могут изменяться от 1 до 10 включительно;

– двумерный массив вещественных чисел A , значения первого и второго индекса могут изменяться соответственно от 1 до 5 и от 1 до 3 включительно.

Обращение к элементу двумерного массива имеет вид:

<Имя массива> (<индекс1> , <индекс2>)

Индексы могут представлять собой выражения целого типа. Тип индекса определяет границы изменения значений индекса.

Например:

➤ $A(3, 2)$ – обращение к элементу третьей строки и второго столбца;

➤ $A(N - 1, M)$ – обращение к последнему элементу предпоследней строки матрицы, состоящей из N строк и M столбцов;

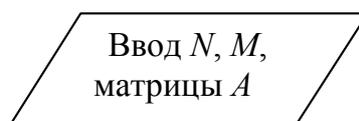
➤ $A(2, j)$ – обращение к элементу второй строки матрицы, при этом номер столбца j должен быть предварительно задан.

2.2. Ввод элементов двумерного массива

При объявлении массива указываются максимально возможные значения индексов, в соответствии с этими значениями в памяти компьютера выделяется память для хранения всех объявленных элементов массива. При выполнении программы можно обрабатывать меньшее количество элементов, задав при вводе матрицы необходимое число строк (N) и столбцов (M) матрицы (размерность $N \times M$).

На рис. 2.2 представлена графическая схема алгоритма и фрагмент программы ввода размерности матрицы (чисел N и M) и элементов матрицы A .

В графических схемах алгоритмов обработки двумерных массивов этот фрагмент оформляется в виде одного блока:



2.3. Вывод элементов двумерного массива

На рис. 2.3 представлена графическая схема алгоритма и фрагменты программы вывода элементов матрицы A в ячейки рабочего листа Excel и в окно сообщений: вариант a и b соответственно. При выводе в окно сообщений процедурой *MsgBox* необходимо предварительно сформировать выводимую строку. Для этого описана переменная *Str* типа *String*, в которой объединяется заголовок массива и все его элементы.

В графической схеме алгоритмов обработки двумерных массивов этот фрагмент оформляется в виде одного блока:

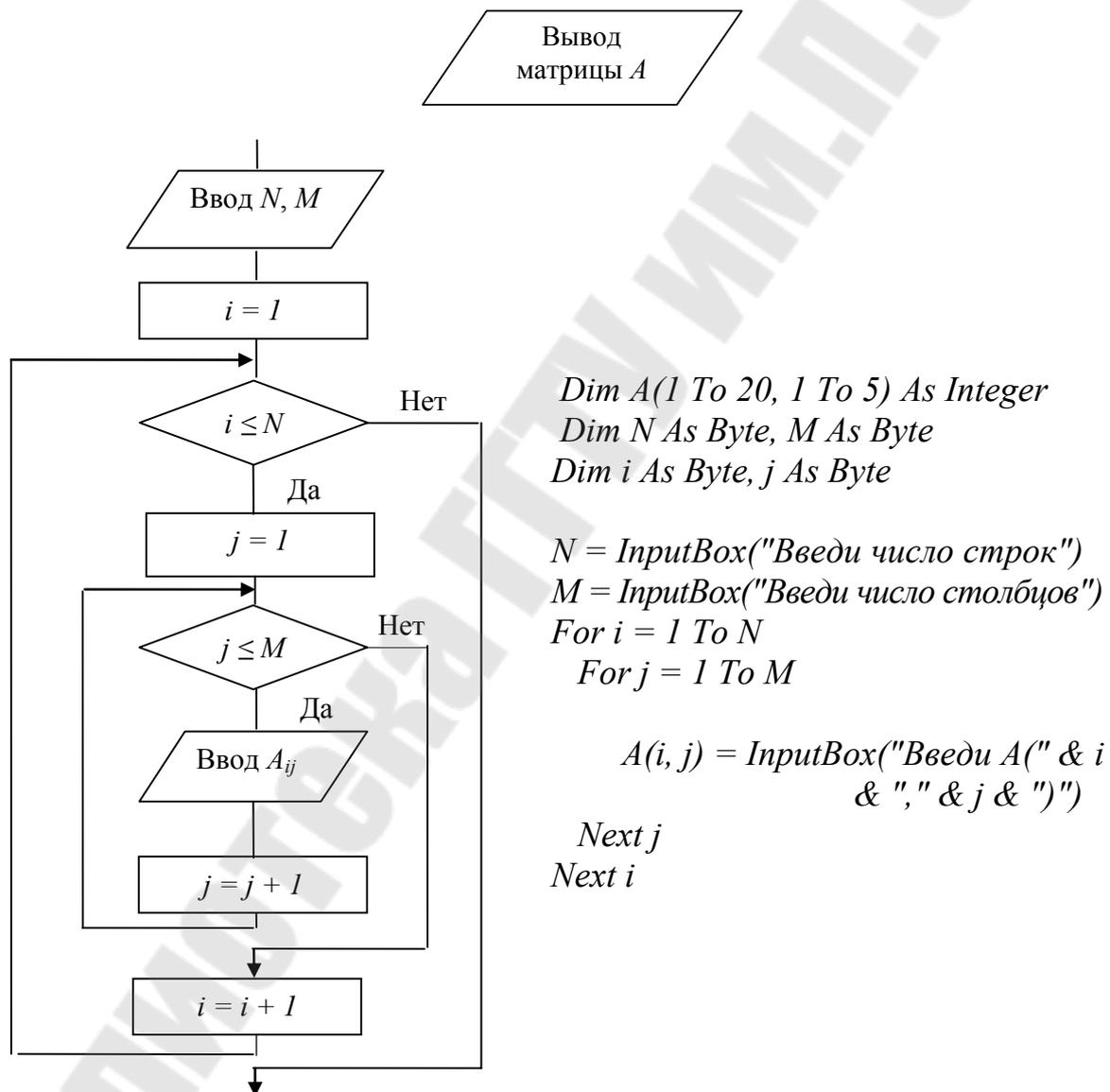
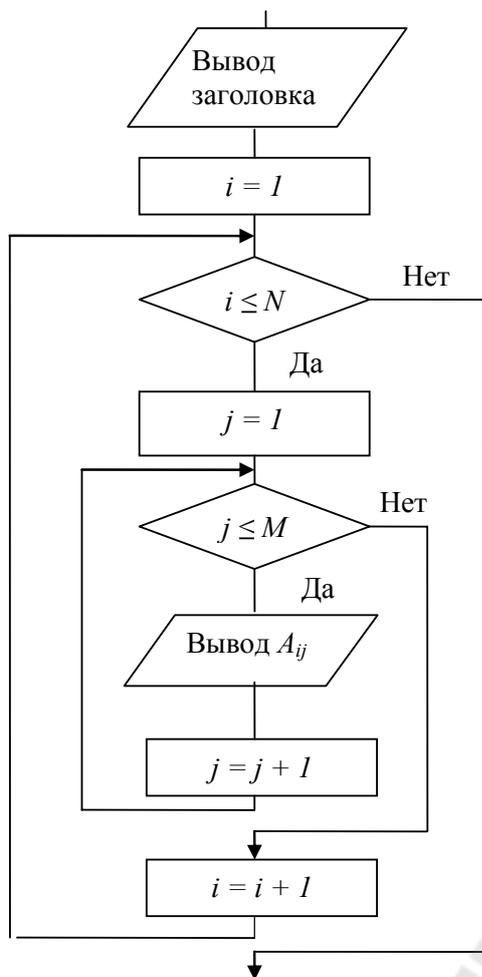


Рис. 2.2. Схема алгоритма и фрагмент программы ввода двумерного массива



а) вывод массива в ячейки рабочего листа:

```

Range("1:5").Clear
Range("A1") = "Исходный массив"
For i = 1 To N
  For j = 1 To M
    Cells(i + 1, j) = A(i, j)
  Next j
Next i
  
```

б) вывод массива в окно сообщений:

```

Dim Str As String
Str = "Исходный массив" & vbCrLf
For i = 1 To N
  For j = 1 To M
    Str = Str & A(i, j) & " "
  Next j
  Str = Str & vbCrLf
Next i
MsgBox Str, , "Вывод массива"
  
```

Рис. 2.3. Схема алгоритма и фрагмент программы вывода элементов двумерного массива

2.4. Типовые алгоритмы обработки двумерных массивов

Формулы для определения суммы, количества, произведения, среднего арифметического элементов матрицы, удовлетворяющих некоторому условию, те же самые, что и при определении соответствующих величин одномерных массивов. Поиск максимальных и минимальных элементов матрицы, как и в одномерном массиве, выполняется в результате сравнения очередного элемента массива с минимальным (максимальным) на данный момент элементом массива.

Отличие при обработке матриц состоит в том, что, во-первых, просматривать матрицу можно по строкам и по столбцам и, во-

вторых, все эти величины можно определять как для всей матрицы, так и для отдельных ее строк и столбцов. Поэтому при определении сумм, количеств, произведений, максимальных и минимальных элементов необходимо следовать следующим правилам:

– для переменных, у которых вычисляется одно значение для всей матрицы, начальное значение устанавливается перед внешним циклом, а результат выводится после его завершения (произведение в примере 2.1, количество столбцов в примере 2.2);

– для переменных, у которых значения зависят от номера столбца или номера строки, начальное значение устанавливается внутри внешнего цикла перед внутренним, а результат выводится или используется после завершения внутреннего цикла в теле внешнего цикла (сумма элементов столбца в примере 2.2);

– если значения переменной вычисляются для каждого или некоторых столбцов, то внешний цикл формируется по номеру столбца; если для каждой или некоторых строк, то внешний цикл формируется по номеру строки (в примере 2.2 вычисляется количество положительных элементов каждого столбца, поэтому внешний цикл организуется по номеру столбца);

– если вычисляется значение переменной для одной строки или одного столбца матрицы, то вложенные циклы не нужны; формируется один цикл по номеру строки или номеру столбца, а значение второго индекса устанавливается постоянным, исходя из условия задачи (пример 2.3).

Пример 2.1. Вычислить произведение элементов матрицы, не принадлежащих интервалу (x,y) и расположенных в столбцах с номерами, кратными 3. Схема алгоритма приведена на рис. 2.4, алгоритм реализован в процедуре *Pr_2_1*.

Пример 2.2. Найти количество столбцов матрицы, в которых сумма положительных элементов больше 20. Схема алгоритма представлена на рис. 2.5, алгоритм реализован в процедуре *Pr_2_2*.

Пример 2.3. Определить максимальный элемент последней строки матрицы и номер столбца, в котором он расположен. Схема алгоритма представлена на рис. 2.6, алгоритм реализован в процедуре *Pr_2_3*.

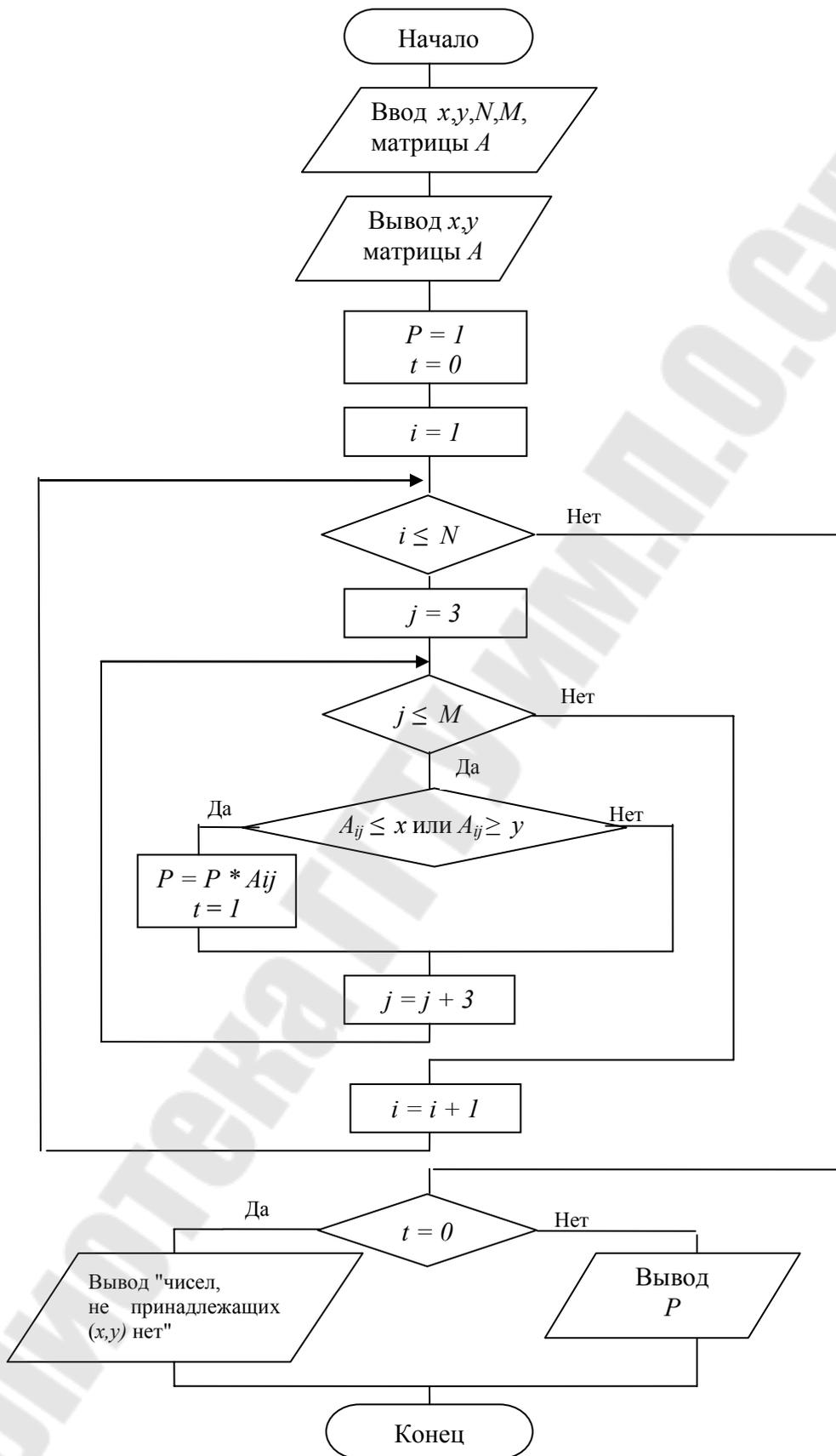


Рис. 2.4. Схема алгоритма примера 2.1

Текст процедуры на VBA:

```
Sub Pr_2_1()  
  Dim A(1 To 20, 1 To 5) As Single, N As Byte, M As Byte  
  Dim i As Byte, j As Byte  
  Dim P As Single, t As Byte  
  N=InputBox("Введите число строк")  
  M = InputBox("Введите число столбцов")  
  For i = 1 To N  
    For j = 1 To M  
      A(i, j) = InputBox("Введи A(" & i & ", " & j & ")")  
    Next j  
  Next i  
  Range("1:10").Clear  
  Range("A1") = "Исходный массив"  
  For i = 1 To N  
    For j = 1 To M  
      Cells(i + 1, j) = A(i, j)  
    Next j  
  Next i  
  Cells(N+2,1) = " Нижняя граница интервала x=" & x  
  Cells(N+3,1) = "Верхняя граница интервала y=" & y  
  P = 1: t = 0  
  For i = 1 To N  
    For j = 3 To M Step 3  
      If A(i,j) <= x or A(i,j) >= y Then  
        P = P * A(i,j)  
        t = 1  
      End If  
    Next j  
  Next i  
  If t = 0 Then  
    Cells(N+4,1) = "Чисел, не принадлежащих (x,y), нет"  
  Else  
    Cells(N+4,1) = "Произведение чисел, не принадлежащих (x,y)=" & P  
  End If  
End Sub
```

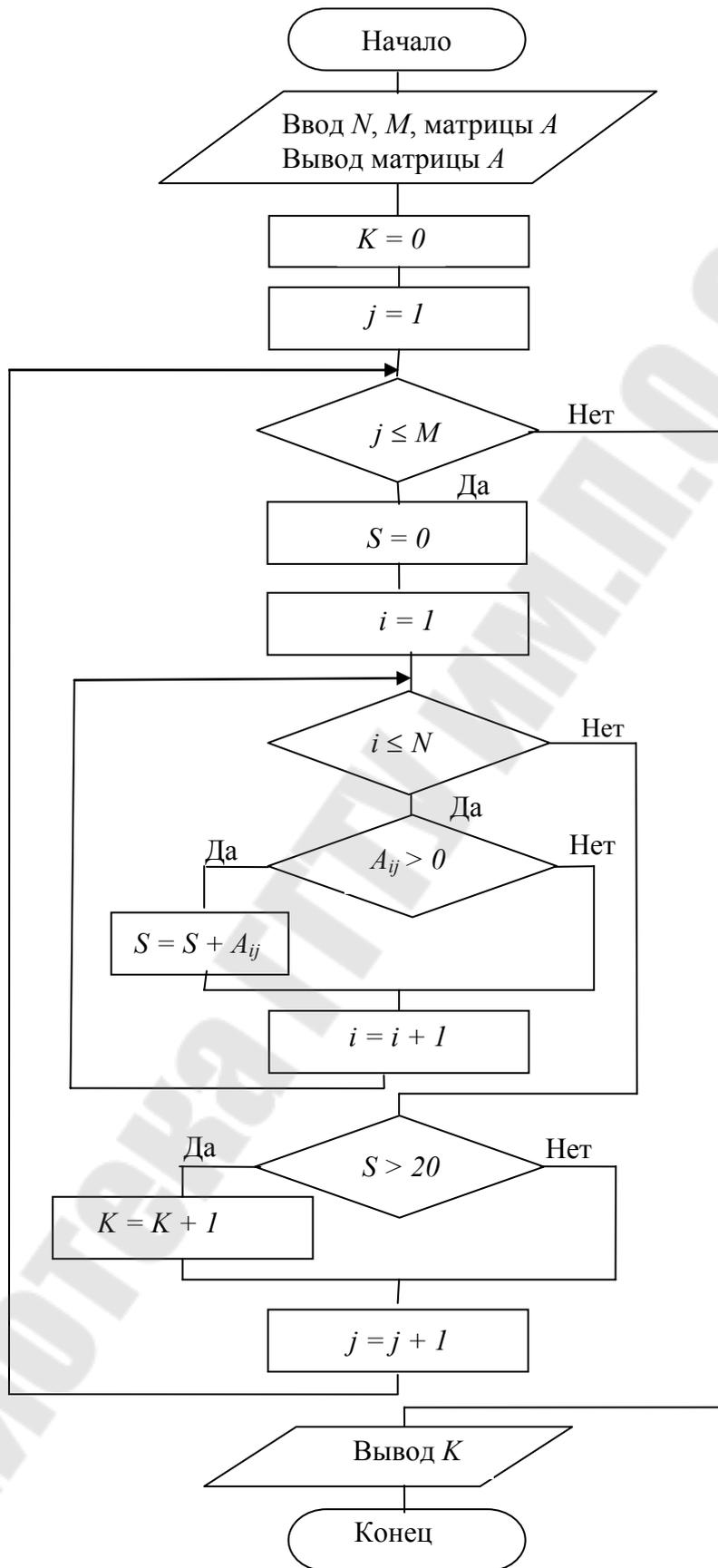


Рис. 2.5. Схема алгоритма примера 2.2

Текст процедуры на VBA:

```
Sub Pr_2_2()  
  Dim A(1 To 20, 1 To 5) As Single  
  Dim N As Byte, M As Byte  
  Dim i As Byte, j As Byte  
  Dim K As Byte, S As Single  
  N = InputBox("Введите число строк")  
  M = InputBox("Введите число столбцов")  
  For i = 1 To N  
    For j = 1 To M  
      A(i, j) = InputBox("Введи A(" & i & ", " & j & ")")  
    Next j  
  Next i  
  Range("1:10").Clear  
  Range("A1") = "Исходный массив"  
  For i = 1 To N  
    For j = 1 To M  
      Cells(i + 1, j) = A(i, j)  
    Next j  
  Next i  
  K = 0  
  For j = 1 To M  
    S = 0  
    For i = 1 To N  
      If A(i, j) > 0 Then S = S + A(i, j)  
    Next i  
    If S > 20 Then K = K + 1  
  Next j  
  If K = 0 Then  
    Cells(N + 2, 1) = "Нет столбцов с суммой положит. элементов > 20"  
  Else  
    Cells(N + 2, 1) = "Количество столбцов с суммой положит. эл-в > 20=" & K  
  End If  
End Sub
```

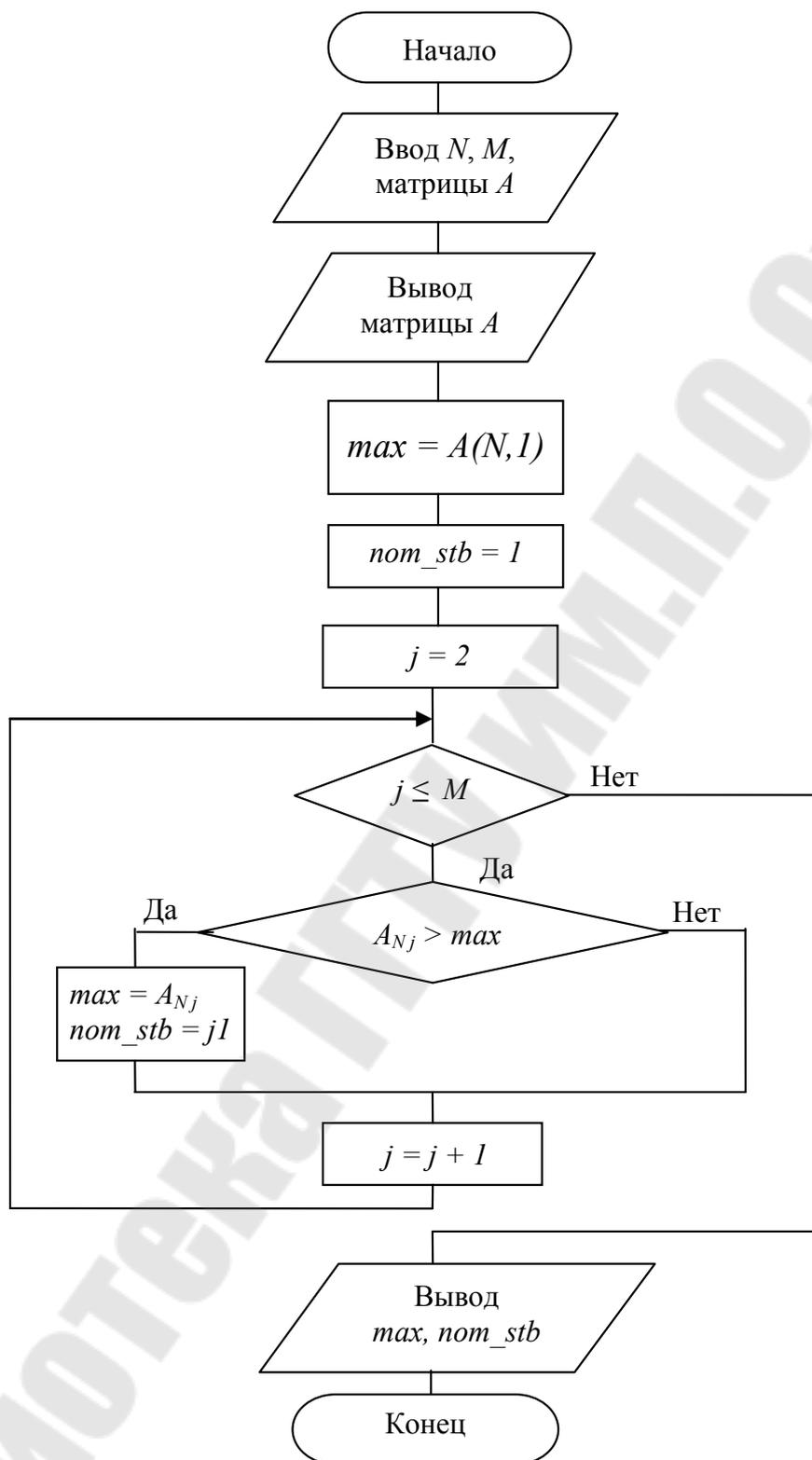


Рис. 2.6. Схема алгоритма примера 2.3

Текст процедуры на VBA:

```
Sub Pr_2_3()  
  Dim A(1 To 20, 1 To 5) As Integer  
  Dim N As Byte, M As Byte, max As Integer  
  Dim i As Byte, j As Byte, nom_stb As Byte  
  N = InputBox("Введите число строк")  
  M = InputBox("Введите число столбцов")  
  For i = 1 To N  
    For j = 1 To M  
      A(i, j) = InputBox("Введи A(" & i & ", " & j & ")")  
    Next j  
  Next i  
  Range("1:10").Clear  
  Range("A1") = "Исходный массив"  
  For i = 1 To N  
    For j = 1 To M  
      Cells(i + 1, j) = A(i, j)  
    Next j  
  Next i  
  max = A(N,1): nom_stb = 1  
  For j = 2 To M  
    If A(N,j) > max Then  
      max = A(N,j)  
      nom_stb = j  
    End If  
  Next j  
  Cells(N+2,1) = "Максимальный элемент последней строки =" & max  
  Cells(N+3,1) = "он находится в столбце с номером " & nom_stb  
End Sub
```

Литература

1. Кузьменко, В. Г. VBA 2002 / В. Г. Кузьменко. – Москва : БИНОМ, 2002. – 624 с.
2. Браун, С. Visual Basic 6 : учеб. курс / С. Браун. – Санкт-Петербург : Питер, 2002.
3. Гарпаев, А. Ю. Excel, VBA, Internet в экономике и финансах / А. Ю. Гарпаев. – Санкт-Петербург : БХВ-Петербург, 2002. – 816 с.
4. Васильев, А. VBA в Office2000 : учеб. курс / А. Васильев, А. Андреев. – Санкт-Петербург : Питер, 2001. – 432 с.
5. Водополова, Н. В. Основы алгоритмизации : практ. пособие / Н. В. Водополова, В. И. Мисюткин, С. А. Чабуркина. – Гомель : ГГТУ им. П. О. Сухого, 2004.
6. Водополова, Н. В. Основы программирования на VBA : практ. пособие / Н. В. Водополова, С. А. Чабуркина. – Гомель : ГГТУ им. П. О. Сухого, 2004.

Содержание

1. Организация работы с одномерными массивами	3
1.1. Объявление одномерного массива	3
1.2. Ввод элементов массива	4
1.3. Вывод элементов массива	6
1.4. Типовые алгоритмы обработки одномерных массивов.....	7
1.4.1. Определение количества, суммы, среднего арифметического элементов массива	8
1.4.2. Определение произведения элементов массива.....	10
1.4.3. Поиск экстремумов.....	13
1.4.4. Формирование нового массива.....	16
1.4.5. Поиск информации в массиве	18
2. Организация работы с двумерными массивами	21
2.1. Объявление двумерного массива.....	21
2.2. Ввод элементов двумерного массива	22
2.3. Вывод элементов двумерного массива	23
2.4. Типовые алгоритмы обработки двумерных массивов	24
Литература	32

Учебное электронное издание комбинированного распространения

Учебное издание

ОБРАБОТКА ОДНОМЕРНЫХ И ДВУХМЕРНЫХ МАССИВОВ НА VBA

Пособие

**по выполнению контрольных и лабораторных
работ по дисциплинам «Информатика» и «Основы
информатики и вычислительной техники»
для студентов экономических специальностей
дневной и заочной форм обучения**

Электронный аналог печатного издания

Авторы-составители: **Водополова** Наталия Виталиевна
Чабуркина София Абелевна

Редактор *Н. В. Гладкова*
Компьютерная верстка *Н. Б. Козловская*

Подписано в печать 29.10.07.

Формат 60x84/16. Бумага офсетная. Гарнитура «Таймс».

Ризография. Усл. печ. л. 2,09. Уч.-изд. л. 1,8.

Изд. № 99.

E-mail: ic@gstu.gomel.by

<http://www.gstu.gomel.by>

Издатель и полиграфическое исполнение:
Издательский центр учреждения образования
«Гомельский государственный технический университет
имени П. О. Сухого».

ЛИ № 02330/0131916 от 30.04.2004 г.

246746, г. Гомель, пр. Октября, 48.