Министерство образования Республики Беларусь

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

Кафедра «Информационные технологии»

В. И. Мисюткин

ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ VBA

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к контрольным работам по курсам «Информатика», «Основы информатики и вычислительной техники» для студентов экономических специальностей заочной формы обучения

Электронный аналог печатного издания

Гомель 2008

УДК 004.43(075.8)
ББК 32.973.26-018.1я73
М65

Рекомендовано к изданию научно-методическим советом факультета автоматизированных и информационных систем ГГТУ им. П. О. Сухого
(*протокол № 7 от 12.03.2007 г.*)

Рецензент: зав. каф. информационно-вычислительных систем БТЭУ ПК
С. М. Мовшович

Мисюткин, В. И.

М65 Программирование на языке VBA : метод. указания к контрол. работам по курсам «Информатика», «Основы информатики и вычислительной техники» для студентов экон. специальностей заоч. формы обучения / В. И. Мисюткин. – Гомель : ГГТУ им. П. О. Сухого, 2008. – 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 978-985-420-753-7.

Рассмотрены основные операторы и конструкции объектно-ориентированного языка программирования VBA. Приводятся примеры программ, разработанные для типовых алгоритмов различных типов: линейных, разветвляющихся, циклических. Данные примеры соответствуют заданиям, которые предлагаются студентам-заочникам при выполнении контрольной работы.

Для студентов экономических специальностей заочной формы обучения.

УДК 004.43(075.8)
ББК 32.973.26-018.1я73

ISBN 978-985-420-753-7

1. Правила оформления контрольной работы

Для выполнения контрольной работы студент должен изучить основы алгоритмизации и программирования на языке VBA. В контрольной работе выполняются задачи, перечень которых определяет преподаватель.

Условия задач выбираются в соответствии с заданным вариантом, номер которого является порядковым номером студента в журнале группы или, в особых случаях, предлагается преподавателем. Список заданий содержится в [5], а в электронном варианте находится на сервере кафедры «Информационные технологии» и может быть просмотрен с любого компьютера, подключенного к кафедральной компьютерной сети.

Для каждой задачи студент должен выполнить и отобразить в контрольной работе:

1. Условие задачи в соответствии с заданным вариантом.
2. Схему алгоритма.
3. Таблицу соответствия переменных (то есть их обозначение в схеме и программе).
4. Рукописный текст программы, написанной на языке VBA.
5. Набор тестов для отладки программы.
6. Листинг (распечатку) текста программы. Листинги *должны* содержать комментарий с фамилией, именем, отчеством и группой студента.
7. Листинг с результатами тестирования программы. *Кроме результатов тестирования он должен содержать все значения вводимых исходных данных*.

Контрольная работа выполняется в отдельной тетради (лучше в клетку) или на листах формата А4 (текст пишется только с одной стороны, все листы брошюруются в папку). Обязательно наличие полей для замечаний проверяющего преподавателя. Листинги вставляются после каждой задачи. Текст оформляется чернилами синего, фиолетового или черного цвета. Схему алгоритма лучше изображать в карандаше и с использованием линейки. При ее оформлении нужно руководствоваться ГОСТ 19.701–90 и методическими указаниями [2], в которых рассматриваются основные типы алгоритмов и их схемы. В конце контрольной работы приводится список литературы, которая была использована при подготовке работы, и студент проставляет свою подпись.

Если контрольная работа возвращена студенту на доработку, то он(а) должен **в той же тетради** выполнить работу над ошибками, причем переписывать всю задачу заново нужно только тогда, когда в ней обнаружены серьезные ошибки (например, алгоритм совершенно неверный). После доработки контрольная работа возвращается преподавателю на повторную проверку.

Правильно выполненная работа допускается к защите. В процессе защиты контрольной работы студент демонстрирует работу составленных им программ путем выполнения их на компьютере. Без выполненной и защищенной контрольной работы студент к экзамену ***не допускается.***

2. Теоретические сведения. Язык Visual Basic for Applications

Назначение: VBA используется в качестве языка программирования в приложениях MS Qffice (Word, Excel, Access, Power Point, и др.). Он позволяет автоматизировать процессы обработки информации и разрабатывать свои сложные приложения, обеспечивающие удобства работы с данными. Это объектно-ориентированный язык. Объекты – это заранее разработанные программы, которые пользователь может использовать в своих программах, не программируя их заново. Он же может разрабатывать и новые объекты. Для работы со средой VBA нужно запустить основное приложение host-Приложение (например, Excel).

Данные и их типы

Данные делятся на константы и переменные.

Типы данных: дата, числа, логический, текстовый, Variant, Object.

Числовые типы: byte, integer, long, single, double, currency.

Integer – целое число (со знаком «+» или «–»). Примеры: 1; 1566; –77. Диапазон: – 32768… +32767 (число занимает 2 байта памяти).

Byte – тоже целые, но из диапазона 0…225 (занимает 1 байт).

Long – длинное целое, имеющее увеличенный диапазон (свыше двух миллионов), поэтому занимает 4 байта памяти.

Single – число одинарной точности (имеет 7 верных значащих цифр); это число с целой и дробной частью (занимает 4 байта памяти). Диапазон: –3,4×103…+3,4×1038. Примеры: 3,458; –100,876.

Экспоненциальная форма записи вещественных чисел:

7,5E11 → 750000000000 (в обычной форме);

–3,2E-8 → 0,0000000032 (в обычной форме).

String – текстовый (строковый) тип (для строк переменной длины занимает: 10 байт + длина строки, которая вычисляется из расчета: 1 байт=1 символ).

Примеры: "Это строка символов"; "5.1".

Boolean – логический тип (занимает 2 байта). Может принимать значения: **true** или **false**.

Правила создания имен переменных и констант:

1. Имя состоит из букв (русских или латинских), цифр и знака подчеркивания "_", набранных любым шрифтом и размером.

2. Первый символ имени – буква.

3. Нельзя использовать в именах: пробел, точку или другие символы, не указанные в п. 1.

4. Нельзя использовать ключевые слова VBA (Sub, Dim, If, For и др.).

5. Имя должно быть уникальным в области свого действия.

6. Желательно, чтобы имя имело содержательный смысл.

Примеры правильных имен: x: z1; Сумма_прибыли, Задание_1.

Неправильные: 1N, Dim, Задание 4.

Объявление переменных и констант

Переменные и константы, используемые в программе, должны быть объявлены (описаны) с помощью конструкции, имеющей следующий синтаксис:

Dim <имя1 > As <тип1 > [, <имя2> As <тип2>]…

где <имяn> – имя переменной, <типn> – ее тип. Скобки [ ] означают – то, что в них заключено, является необязательным элементом.

Примеры:

Dim N As integer, Сумма As Single, B As Boolean

Это типизированные переменные. Они в программе должны принимать значения, соответствующие указанному для них типу. Хотя в ряде случаев VBA способен преобразовать один тип данных в другой.

Пример описания строковой переменной фиксированной длины (40 символов):

Dim строка As String*40.

Область действия переменных: процедура или модуль, в которых они описаны.

Именованные константы

Их имена формируются по тем же правилам, что и имена переменных, а значения задаются при описании, но изменяться в программе не могут. Оператор для описания имеет вид:

Const <имя1> = < значение1]> <[, < имя2>=< значение2> …

Пример: Const Pi=3.14159: G=9.81:St="ГГТУ имени П. О. Сухого"

Структура программы (процедуры)

Рассмотрим структуру программы на простом примере (рис. 2.1):

Заголовок ⟶	Sub Pr()
Комментарий ⟶	'Выполнил студент Петров, гр.ЗО-11
< раздел описаний >	Dim a As Single,b As Single,z As Single
< тело процедуры >	z=a+b:
	MsgBox z
Конец процедуры ⟶	End Sub

Рис. 2.1

Допускается запись нескольких операторов в одной строке, разделяя их символом ":". Но длина строки не должна превышать 1024 символов. Не делайте слишком длинные строки, так как могут возникнуть проблемы с распечаткой текста программы на принтере или при просмотре (оптимально до 80 символов). Длинную строку можно отобразить на нескольких строках, используя в качестве символа переноса комбинацию пробела со знаком подчеркивания " _".

Текст программы желательно набирать с отступами с левой стороны, с помощью которых выделяются отдельные части программы, и она становится более удобной для просмотра.

Программа может включать комментарии, поясняющие выполняемые ею действия. Комментарий начинается со знака ' (апостроф) и может содержать в себе текст с любыми символами. Он формируется в виде отдельной строки или вместе с одним из операторов. На цветном мониторе редактор VBA отображает комментарии зеленым цветом, ключевые слова VBA – синим, а остальной текст программы – черным.

Примеры использования комментариев:

' программу составил ст. гр.ЗО-11 Иванов И.И.

d=InputBox("Введи d","Окно ввода") ' Ввод исходного значения d

Оператор присваивания (=)

Используется для присваивания значений переменным и имеет следующий синтаксис:

< имя переменной > = < выражение >.

Типы левой и правой частей должны совпадать. В ряде случаев VBA автоматически осуществляет преобразование, а когда это невоз-

можно происходит остановка выполнения программы (Run-time error – ошибка выполнения).

Ввод данных

Чтобы ввести значение переменной, нужно использовать оператор, имеющий следующую структуру:

< имя переменной > = InputBox (< строка > [, < заголовок >]),

где InputBox – функция, которая используется для ввода данных, <строка> – текстовая подсказка, в которой говорится о том, что требуется ввести; <заголовок> – необязательный параметр, который является заголовком окна, открывающегося при работе функции InputBox (если он отсутствует, то окно будет названо именем host-Приложения – MSExcel.

Пример:

d = Input Box ("Введите количество дней" ","Ввод данных")

При работе оператора мы увидим окно следующего вида, в текстовое поле которого нужно ввести значение d и нажать на кнопку ОК (рис. 2.2).

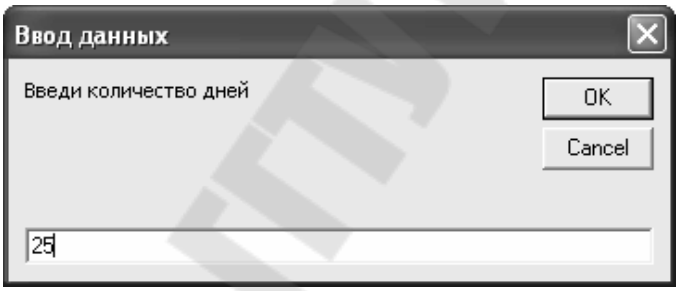


Рис. 2.2

Примечание. При вводе вещественных чисел, имеющих дробную часть, нужно обе части числа разделять символом ",", а не "."!

Вывод данных

Для этого используется оператор, имеющий следующую структуру:

MsgBox < строка > [,, < заголовок >],

где MsgBox – процедура, используемая для вывода данных. Выводимое значение должно быть представлено в виде <строки>; чаще всего VBA преобразует данные в строку символов сам, либо это делается с помощью специальных функций преобразования; <заголовок> имеет тот же смысл, что и у функции InputBox. Отдельные строки при выводе можно сцепить при помощи операции конкатенации (&).

Примечание 1. Две запятые обязательны! (здесь пропущен необязательный параметр).

Примеры:

1) MsgBox "Привет, студент", , "Приветствие"

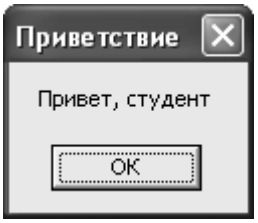


Рис. 2.3

Окно появляется в host-Приложении. После его просмотра нужно нажать на кнопку ОК.

2) K=5
Z=7.3
MsgBox "K=" & K & " Z=" & Z, , " Результаты"

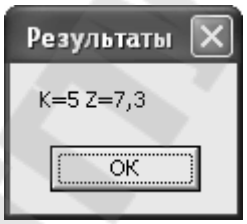


Рис. 2.4

Примечание 2. Для вывода данных в несколько строк в выводимую строку нужно включать функцию Chr(13). Сделаем это для примера 2

MsgBox "K=" & K & Chr(13) & "Z=" & Z, ,"Результаты"
и получим

Рис. 2.5

Для распечатки этого окна необходимо:

Скопировать данное активное окно в буфер обмена (нажав вместе клавиши <Alt> - <PrintSreen>).

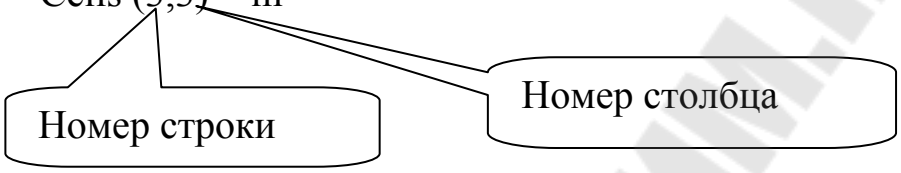
Нажать OK.

Вернуться в Excel.

Вставить содержимое буфера на лист (командой Правка → Вставить).

Вывод в ячейки листа рабочей книги. Для этого используется свойство Cells объекта Range

M=5.37
Cells (3,2) = “Результат m=”
Cells (3,3) = m



В результате выполнения этих операторов на листе рабочей книги мы увидим (для этого нужно переключиться на Excel):

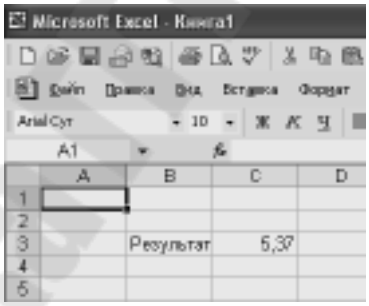


Рис. 2.6

Выражения

Они бывают: численные, строковые и логические.

Нельзя использовать несовместимые типы в одном выражении (кроме числовых).

Арифметические выражения могут содержать: константы, переменные, имена функций, знаки операций и круглые скобки. Результат вычисления арифметического выражения – число.

Перечислим операции в порядке их выполнения в одном выражении (приоритета):

^ (степень)
– (унарный минус)
* и / (умножение и деление)
\ (целочисленное деление)
mod (деление по модулю)
+ и – (сложение и вычитание).

Операции, имеющие одинаковый приоритет (например: * и /, + и -), выполняются слева направо, а приоритет операции можно повысить, заключив ее в круглые скобки.

Функции имеют самый высокий приоритет и записываются с аргументом, заключенным в круглые скобки. Перечень основных функций: Abs(x) – модуль числа, Ant(x) – арктангенс, Cos(x) – косинус, Sin(x) – синус, Tan(x) –- тангенс, Exp(x) – экспонента (число е в степени x), Log(x) – натуральный логарифм, Sqr(x) – квадратный корень. Аргумент функции – любой численный тип.

Примеры:

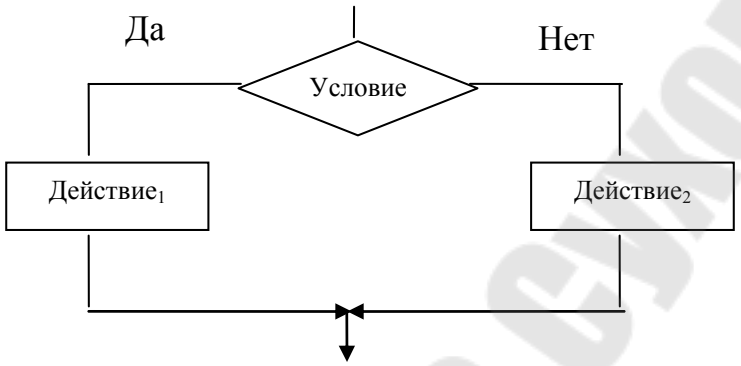
Математическая запись	Запись на VBA
$Z=\sin(\alpha^2)$	Z=Sin(a^2) (Здесь α - это a)
$b=\cos^2(y)$	b=Cos(y)^2
$\phi = \sqrt[7]{n} + \lg m$	Fi=n^(1/7)+Log(m)/Log(10) (Так записывается десятичный логарифм)
$p = \frac{e + c^{0.7}}{abc}$	P=(Exp(-a)+c^0.7)/(a*b*c)

Рис. 2.7

Условный оператор If

Служит для проверки выполнения условий, которая в схеме алгоритма реализуется с помощью символа Решение и конструкции Ветвление. Существует в двух формах записи: полной и сокращенной (рис. 2.8).

Полная форма
IF <условие> Then
  <Действие1>
Else
  <Действие2>
EndIf



Сокращенная форма
If <условие> Then
 <Действие>
End IF

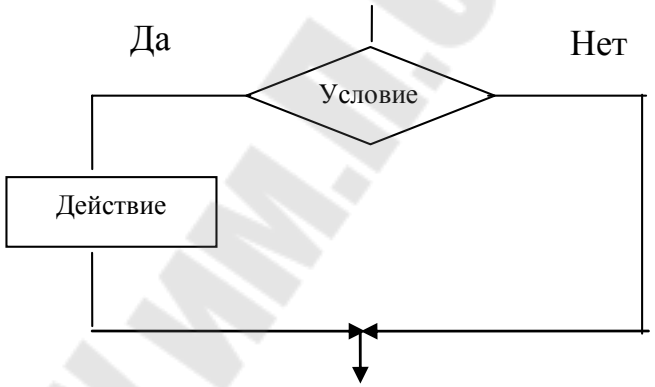


Рис. 2.8

В записи оператора <условие> представлено в виде логического выражения, например, Z>5, и может принимать значения True (ИСТИНА) и False (ЛОЖЬ).

<Действие> представляет собой один или несколько операторов. Действие, записанное в ветви ДА, выполняется тогда, когда условие имеет значение True, а когда значение условия False, выполняется Действие, записанное в ветви Нет (полная форма) или ничего не выполняется (сокращенная форма).

Пример

$Y = \begin{cases} \sqrt{x}\ , & \text{если} \quad x > 0\ , \\ x^2\ , & \text{если} \quad x \le 0 \end{cases}$	If X>0 Then
	  Y=Sqr(X)
	Else
	  Y=X^2
	Endif

Рис. 2.9

Имеют также место операторы If, вложенные друг в друга.

Примечание. Существует запись оператора If в одну строку, но она при всей своей компактности делает программу менее читаемой, поэтому здесь она не приводится.

Логические выражения могут включать в себя следующие элементы: логические переменные, логические константы, операции сравнения и логические операции.

Операции сравнения: < , >, < = , >= , = , < >.

Логические операции: And (умножение), Or (сложение), Not (отрицание).

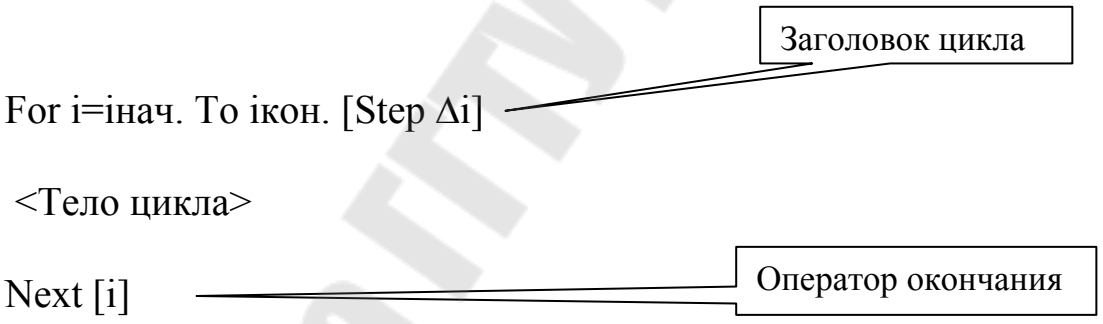
Примеры:

а) 1<x≤1001 x>0 And x<=1001

б) y<0 или y>10 y<0 Or y>10

Оператор цикла For…Next.

Цикл – это многократное повторение одной и той же группы операторов. Для его организации в VBA имеется специальный оператор со следующей синтаксической структурой:



For (для), To (до), Next (следующий) – служебные слова, i – параметр цикла, iнач., iкон. – его начальное и конечное значения, Δi – шаг изменения (может быть как положительным, так и отрицательным, а если он отсутствует, то полагается равным 1).

Все перечисленные параметры имеют одинаковый тип (чаще всего Integer или Single). Конец цикла Next может быть записан без указания параметра i.

Такой цикл реализует структуру цикла с предусловием (где сначала проверяется, можно ли выполнить цикл и только тогда, когда можно – тело цикла выполняется).

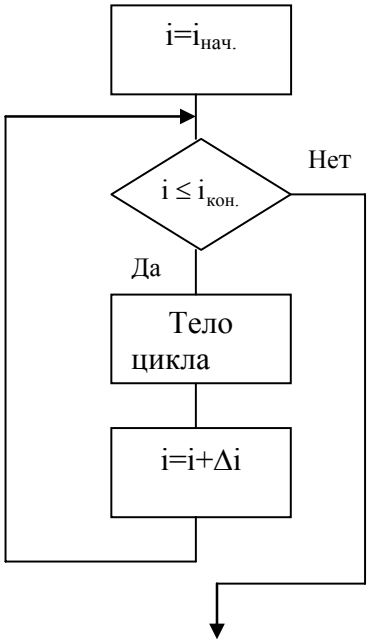


Рис. 2.10

Цикл работает по приведенной здесь схеме следующим образом:

а) при Δi>0 – параметр i последовательно пробегает все значения от iнач. до iкон. с шагом Δi; при этом должно быть iнач. ≤iкон.;

б) при Δi<0 – параметр i последовательно пробегает все значения от iкон. до iнач. с шагом Δi; при этом должно быть iнач. ≥iкон. Увеличение или уменьшение параметра i на каждом шаге цикла (итерации) происходит автоматически и не требует программирования его изменения.

Примеры

1)
```
For i=1 To 10
  Cells(i,1)=i
Next i
```

Выполнение этого фрагмента приведет к появлению в первых десяти ячейках столбца A рабочего листа последовательности из натуральных чисел 1,2,…, 10.

2)
```
For m=10 To -10 Step -2
  MsgBox m
Next m
```

Выполнение этого фрагмента приведет к появлению последовательности окон, содержащих числа 10,8,…, –10.

Массивы

До сих пор мы имели дело с переменными, принимающими в ходе выполнения программы одно единственное значение. Но, например, в математике мы имеем дело с такими понятиями как вектор, матрица, которые включают в себя целый набор значений (элементов). В языках программирования такие величины объединяют одним понятием – массив. Таким образом, массив – это некоторый ограниченный набор величин одного типа.

Для объявления (описания) массива, позволяющего в памяти компьютера зарезервировать место для хранения значений элементов массива, используется следующий оператор, располагающийся в разделе описаний процедуры:

Dim <имя массива> (измерение(я)) As <тип элементов>, [(измерение(я)) As <тип элементов>], …
где <измерение> задается в виде целого числа, показывающего самое большое значение, которое может принять данное измерение.

Чтобы взять из массива нужную величину, необходимо указать ее порядковый номер (индекс), который является целым числом (обычно положительным или равным 0). Индексов может быть несколько, и тогда они перечисляются через запятую.

По умолчанию начальное значение индекса равно нулю, а чтобы индекс у элементов массива изменялся, начиная с 1, нужно перед описанием всех переменных, констант и процедур, входящих в модуль, поставить директиву **Option Base** 1.

Примеры объявлений:

Option Base 1

Dim X(5) As Single 'объявлен одномерный массив (вектор) из 5 вещественных чисел

Dim Y(100) As Integer 'то же но из 100 целых

Dim Z(7,6) As String 'двумерный массив (матрица) из 7 строк и 6 столбцов

Ввод одномерного массива

Чтобы заполнить область, отведенную под массив данными, нужно использовать приведенную здесь циклическую структуру. Для определенности будем считать, что нужно ввести массив X, состоящий из N элементов.

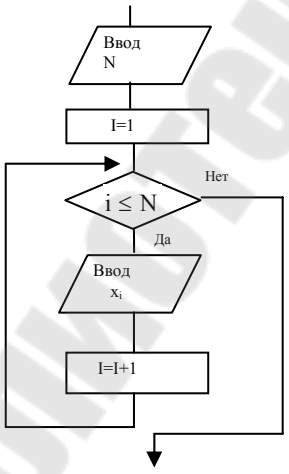
Схема	Программный фрагмент
	…… Dim X(20) As Single 'фрагмент для ввода массива N=InputBox ("Сколько чисел?", "Ввод Размерности") For i=1 to N X(i)=InputBox ("Введи X(" & I & ")","Ввод элементов") Next i ……..

Рис. 2.11

Вывод одномерного массива

Осуществляется практически по той же схеме, что и ввод, с той лишь разницей, что вывод можно осуществить в окно при помощи процедуры MsgBox или в ячейки листа рабочей книги Excel. Во фрагменте программы показано последнее, причем числа выводятся в столбец.

Схема	Программный фрагмент

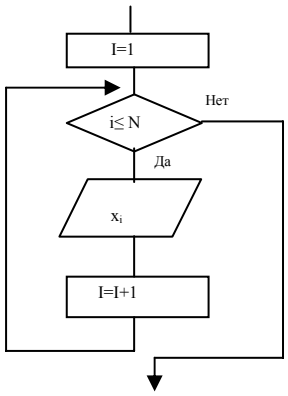
	……..
	' вывод массива
	Cells(1,1)="№ П/П"
	Cells(1,2)="X(i)"
	For i=1 to N
	  Cells(I+1,1)=i
	  Cells(I+1,2)=X(i)
	Next i
	……

Рис. 2.12

Типовые алгоритмы обработки одномерных массивов

Среди множества разнообразных алгоритмов, использующихся для обработки одномерных массивов, чаще всего встречаются следующие типы:

- Вычисление сумм, произведений и количества.
- Определение наименьшего или наибольшего по значению элемента и его порядкового номера.
- Перестановка и замена элементов в массиве.
- Формирование нового массива из элементов имеющихся.

Примеры разработки схем таких алгоритмов и программ, написанных для них, приведены в следующем разделе.

3. Примеры составления программ

3.1. Линейный алгоритм

Пример 3.1

Вычислить $\phi = \sqrt{\alpha} + \beta^2$, при заданных α, β

1. Составляем схему алгоритма (рис. 3.1).

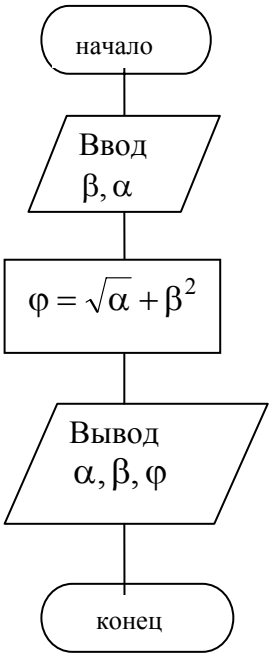


Рис. 3.1. Схема алгоритма примера 3.1

2. Составляем таблицу соответствия переменных:

Таблица 3.1

В схеме	**В программе**	**Тип**	**Комментарий**
φ	F	Single	Результат
α	L	Single	Исходные данные
β	B	Single	Исходные данные

3. Составляем текст программы.

```
Sub Лин_Пример()
'Составил студент группы ЗО-11с Иванов И. И.
Dim L As Single, B As Single, F As Single
L=InputBox ("Введи L", "Ввод")
B=InputBox ("Введи B", "Ввод")
F=Sqr(L)+B^2
MsgBox "При L=" & L & " B=" & B & Chr(13) & " F=" & F
End Sub
```

4. Готовим тесты для проверки правильности работы программы.

Для чего подбираем такие числовые значения исходных данных, для которых результат легко просчитывается устным счетом или с помощью микрокалькулятора (табл. 3.2).

Таблица 3.2

α	β	φ
9	8	67
1,44	0,3	1,29

Результаты тестирования представлены в следующих окнах:

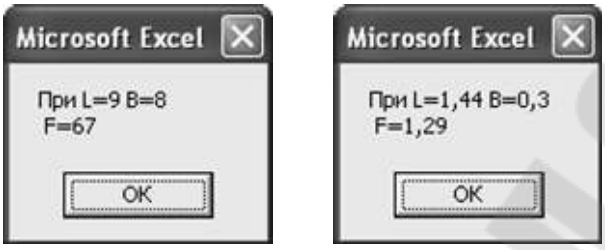


Рис. 3.2

Они доказывают правильность работы программы, так как совпадают с результатами, полученными «ручным» счетом.

3.2. Разветвляющийся алгоритм

Пример 3.2

Составить программу для расчета функции

$$F = \begin{cases} \lg^2 x, \textit{если } 0 < x \le 1000 \\ x^2, \textit{ если } x \le 0 \\ \sqrt[3]{x}, \textit{ в остальных случаях.} \end{cases}$$

Кроме значения F нужно выводить и номер рабочей формулы.

Результаты нужно выводить в ячейки рабочего листа, оформив их в виде следующей таблицы (в ней сразу же приводится полный набор тестов; для этого взято по одной точке из каждого отрезка и точки, в которых меняется формула).

Таблица 3.3

Номер теста	x	F	Номер формулы
1	-2	4	2
2	0	0	2
3	100	4	1
4	1000	9	1
5	8000	20	3

1. Составляем схему алгоритма (рис. 3.3).
2. Составляем таблицу соответствия переменных.

Таблица 3.4

В схеме	**В программе**	**Тип**	**Комментарий**
x	X	Single	Исходные данные
F	F	Single	Результат
n	N	Integer	Результат
–	I	Integer	Вспомогательная переменная
–	K	Integer	Вспомогательная переменная

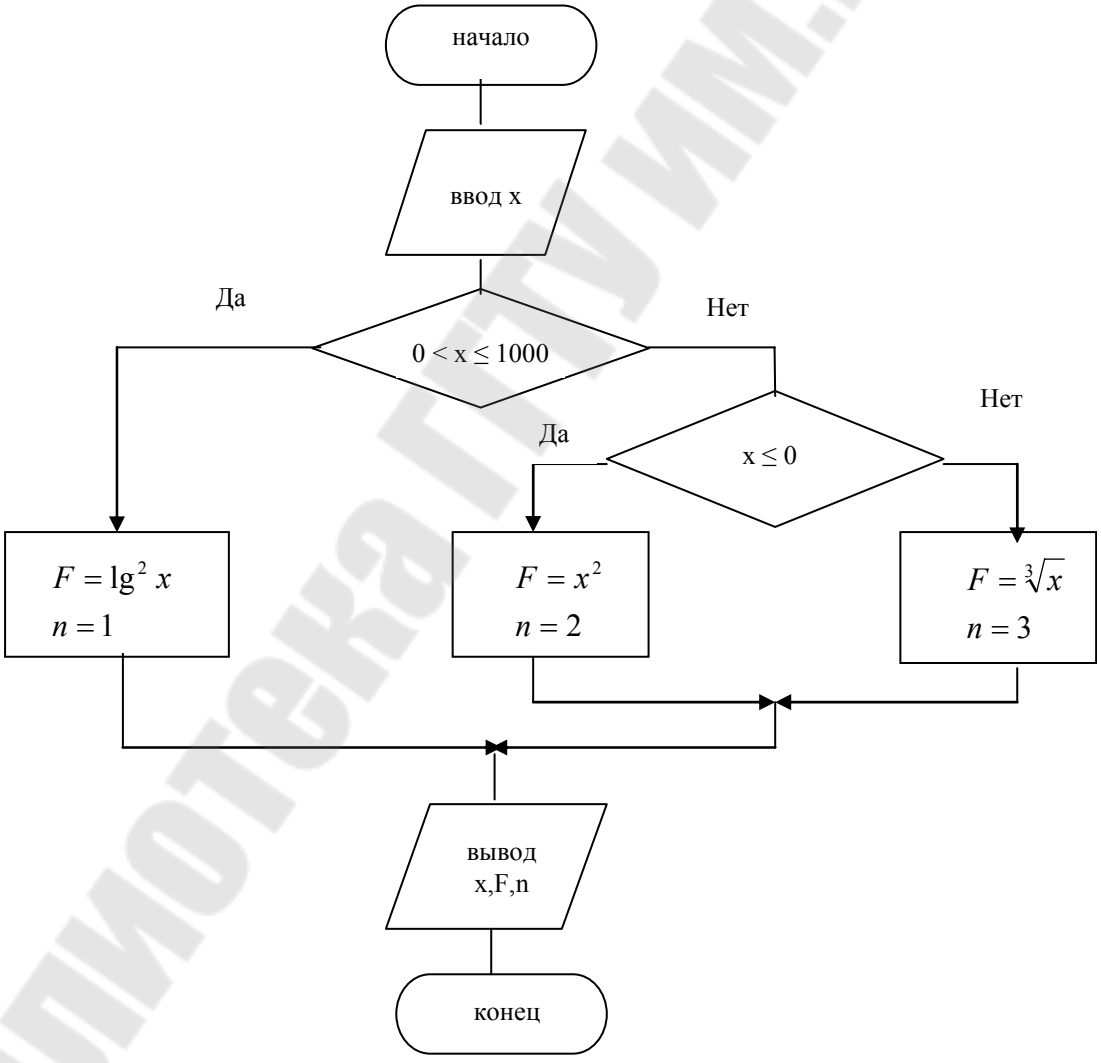


Рис. 3.3. Схема алгоритма примера 3.2

3. Составляем текст программы, в которую добавлен цикл, позволяющий сразу же проводить расчеты для всего набора тестов. Выполнение программы для приведенных выше тестов позволит сделать выводы о правильности ее работы.

```
Sub BETB3()
' Составил студент группы ЗО-11с Петров Т.М.
Dim I As Integer, N As Integer, X As Single, F As Single, K As Integer
K = InputBox("Сколько всего тестов?", "Ввод")
'Формирование заголовка
Cells(1, 1) = "№ теста"
Cells(1, 2) = "X"
Cells(1, 3) = "Y"
Cells(1, 4) = "№ формулы"
For I = 1 To K ' цикл для повторения расчетов
 X = InputBox("Введи X", "Ввод")
 If X > 0 And X <= 1000 Then
   F = (Log(X) / Log(10)) ^ 2
   N = 1
 Else
   If X <= 0 Then
     F = X ^ 2
     N = 2
   Else
     F = X ^ (1 / 3)
     N = 3
   End If
 End If
 Cells(I + 1, 1) = I
 Cells(I + 1, 3) = F
 Cells(I + 1, 4) = N
Next I
End Sub
```

3.3. Циклический алгоритм табулирования функции

Пример 3.3

Составить программу для табулирования функции y=sin x на заданном отрезке [хнач,хкон] с заданным шагом Δx. Результаты нужно представить в виде таблицы на рабочем листе Excel:

Таблица 3.4

№ п/п	x	y
1	…	…
2	…	…
…	…	…

Решение

Поскольку все действия по вычислению функции и записи результатов в таблицу являются одинаковыми для любого значения x, организуем их выполнение в цикле. Параметром цикла будет являться переменная x, принимающая в нем ряд значений хнач, хнач+Δx, хнач+2·Δx, …, хкон. В программу введена дополнительная переменная i, определяющая порядковый номер строки с результатом и номер строки таблицы, в которую он будет выведен (он равен i+1, так как строка 1 занята заголовком).

1. Составляем схему алгоритма (рис. 3.4).
2. Составляем таблицу соответствия переменных:

Таблица 3.5

В схеме	В программе	Тип	Комментарий
x	x	Single	Параметр цикла
хнач	xn	Single	Исходное данное
хкон	xk	Single	Исходное данное
Δx	dx	Single	Исходное данное
y	y	Single	Результат
	i	Integer	Вспомогательная переменная

3. Составляем текст программы

```
Sub Tabl()
'Составил Иванов И.И., гр.ЗО-11
Dim i As Integer, x As Single
Dim xn As Single, k As Single, dx As Single
xn = InputBox("Введи Хнач")
xk = InputBox("Введи X кон")
dx = InputBox("Введи шаг")
'Заголовок таблицы
Cells(1, 1) = "№ п/п"
Cells(1, 2) = "x"
Cells(1, 3) = "y"
Cells(1, 5) = "хнач=" & xn
```

```
Cells(2, 5) = "хкон=" & xk
Cells(3, 5) = "шаг=" & dx
i = 1
For x = xn To xk Step dx
Cells(i + 1, 1) = i
Cells(i + 1, 2) = x
Cells(i + 1, 3) = Sin(x)
i = i + 1
Next x
End Sub
```

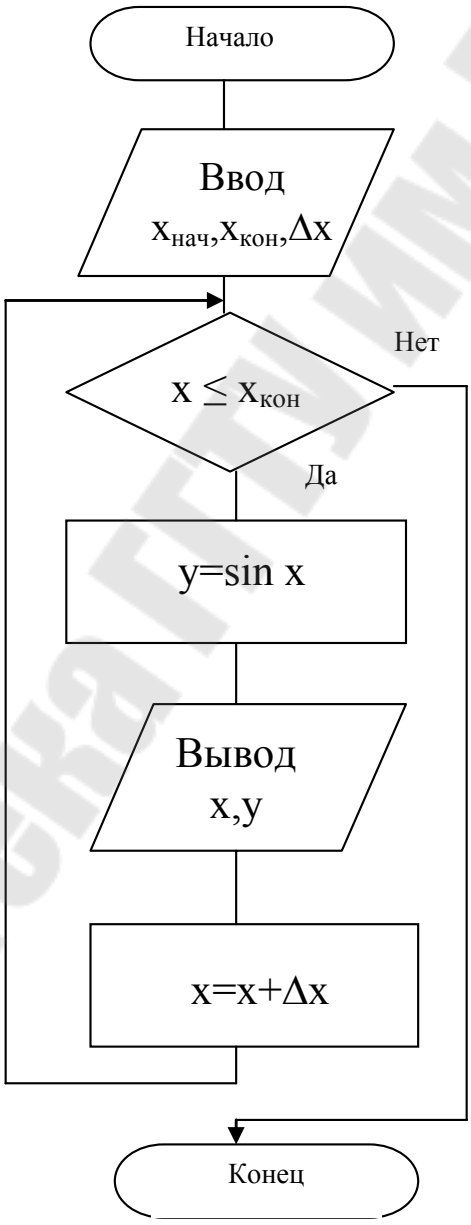


Рис. 3.4. Схема алгоритма примера 3.3

4. Составляем тесты для проверки правильности работы программы:

Таблица 3.5

№ п/п	x	y	хнач=0
1	0	0	хкон=3,14
2	0,314	0,308866	шаг=0,314
3	0,628	–	–
4	0,942	–	–
5	1,256	–	–
6	1,57	–	–
7	1,884	–	–
8	2,198	–	–
9	2,512	–	–
10	2,826	–	–

Обратите внимание! Не имеет смысла вручную просчитывать все значения этой таблицы. Достаточно это сделать в двух любых соседних точках, так как в остальных расчет ведется по той же самой формуле.

3.4. Циклический алгоритм вычисления суммы и произведения элементов массива

Пример 3.4

В массиве X, состоящем из N чисел, вычислить сумму элементов, значения которых принадлежат отрезку (a, b] и произведение положительных элементов, стоящих на чётных местах.

Составим фрагмент схемы алгоритма (**в контрольной работе схему и программу составлять полностью!**) (рис. 3.5).

2. Составляем таблицу соответствия переменных:

Таблица 3.6

В схеме	В программе	Тип	Комментарий
массив X	X	Single	Исходное данное
a	a	Single	Исходное данное
b	b	Single	Исходное данное
P	P	Single	Результат
K	K	Integer	Вспомогательная переменная
S	S	Single	Результат
i	i	Integer	Вспомогательная переменная
N	N	Integer	Исходное данное

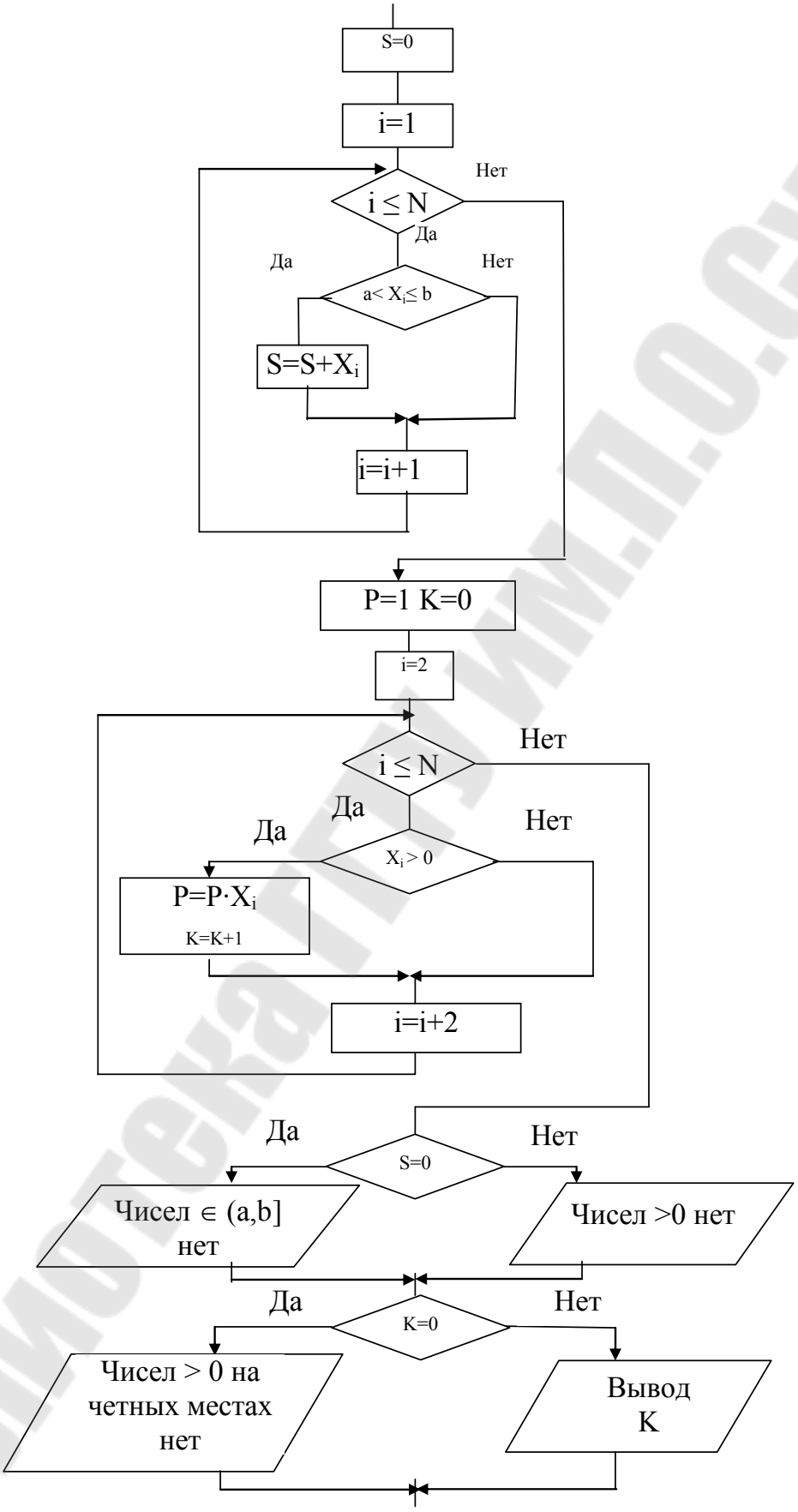


Рис. 3.5. Схема алгоритма примера 3.4

3. Составляем фрагмент программы

```
……..
S =0
For i = 1 To n
  If X(i) > a And X(i) <= b Then
     S = S + X(i)
  End If
Next i
P = 1: K = 0
For i = 2 To n Step 2
  If X(i) > 0 Then
     P = P * X(i)
     K = K + 1
  End If
Next i
'Вывод суммы
If S=0 Then
  MsgBox "чисел, принадлежащих(" & a & ", " & b & "] Нет",, _
"Вывод сообщения"
Else
MsgBox "сумма чисел, принадлежащих (" & a & "," & b & ") = " & _
 S,, "Вывод суммы"
End If
'Вывод произведения
If K = 0 Then
  MsgBox "чисел >0 на чётных местах нет",, "Вывод сообщения"
Else
  MsgBox "Произведение чисел >0 на чётных местах =" & P _
  & Chr(13) & " их количество = " & K,, "Вывод произведения"
End If
```

4. Составляем тесты для проверки правильности работы программы.

При тестировании программы нужно рассмотреть все возможные случаи, которые могут возникнуть для разных исходных данных:

а) при вычислении суммы:

– все числа массива $\in$(a,b];

– все числа массива $\notin$(a,b];

– часть чисел $\in$(a,b], а часть – нет;

б) при вычислении произведения:

– все числа, стоящие на четных местах >0;

– на четных местах нет чисел >0;

– на четных местах часть чисел >0, а часть нет.

Указанные ситуации можно создать, подбирая соответствующим образом числа массива и (или), изменяя значения a и b.

Тест 1

N=6 a=0 b=10

Таблица 3.7

X_1	X_2	X_3	X_4	X_5	X_6
1	4	5	2	7	3

Ожидаемые результаты: S=22 P=24 K=3.

Тест 2

N=6 a=8 b=10

Таблица 3.8

X_1	X_2	X_3	X_4	X_5	X_6
1	-2	3	-4	7	-6

Ожидаемые результаты: чисел $\in$(8,10] нет, чисел >0 на четных местах – нет.

Тест 3

N=6 a=3 b=6

Таблица 3.9

X_1	X_2	X_3	X_4	X_5	X_6
4	3	5	-4	6	5

Ожидаемые результаты: S=20 P=15 K=2.

3.5. Циклический алгоритм нахождения максимального элемента массива и его номера

Пример 3.5

Найти максимальный элемент массива X и поменять его местами с предыдущим.

Обозначим:

– максимальное значение – max,

– номер максимального элемента – n_max,

тогда номер предыдущего элемента будет – n_max-1.

1. Составляем фрагмент схемы алгоритма (рис. 3.6).
2. Составляем таблицу соответствия переменных:

Таблица 3.10

В схеме	В программе	Тип	Комментарий
массив x	x	Single	Исходное данное
max	max	Single	Результат
n_max	n_max	Integer	Результат
N	N	Integer	Исходное данное
i	i	Integer	Вспомогательная переменная

3. Составляем фрагмент программы

```
……….
Cells(1, 1) = "Исходный массив"
For i = 1 To N
 Cells(2, i) = x(i)
Next i
max = x(1)
n_max = 1
For i = 2 To N
  If x(i) > max Then
   max = x(i)
   n_max = i
  End If
Next i
'Перестановка
If n_max = 1 Then
 MsgBox "Перестановка невозможна",, "Сообщение"
 Else
  R = x(n_max)
  x(n_max) = x(n_max - 1)
  x(n_max - 1) = R
 End If
Cells(3, 1) = "Результирующий массив"
For i = 1 To N
 Cells(4, i) = x(i)
Next i
……
```

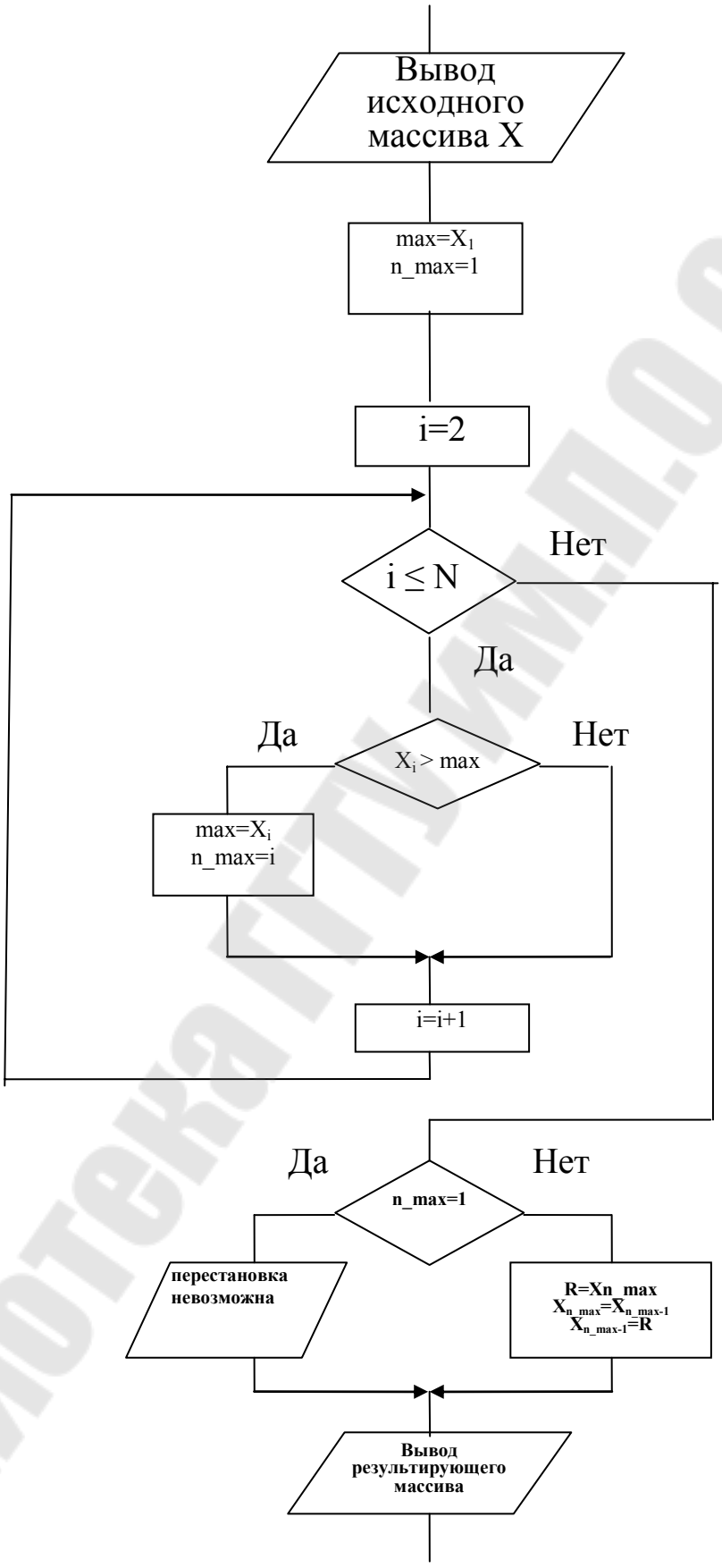


Рис. 3.6. Схема алгоритма примера 3.5

4. Составляем тесты для отладки программы.

Тест 1

Максимальный элемент стоит на произвольном месте

N=4. Исходный массив:

Таблица 3.11

X_1	X_2	X_3	X_4
-7	8	12	4

Ожидаемые результаты:

max=12 n_max=3 Результирующий массив:

Таблица 3.12

X_1	X_2	X_3	X_4
-7	8	12	4

Тест 2

Максимальный элемент стоит первым

N=3. Исходный массив:

Таблица 3.13

X_1	X_2	X_3
20	8	1

Ожидаемые результаты:

max=20 n_max=1. Перестановка невозможна, массив не изменяется

3.6. Циклический алгоритм формирования нового массива из элементов имеющихся массивов

Пример 3.6

Даны два массива A и B, состоящие из N и M чисел, соответственно. Из отрицательных элементов массивов обоих массивов сформировать новый массив C.

Введем обозначения:

– k – количество элементов в массиве C (он же – индекс у вновь создаваемых элементов),

– i – счетчик просмотренных элементов в массивах A и B (индекс у просматриваемых элементов).

1. Составляем фрагмент схемы алгоритма (рис. 3.7).

2. Составляем таблицу соответствия переменных:

Таблица 3.14

В схеме	**В программе**	**Тип**	**Комментарий**
массив A	A	Single	Исходное данное
Массив B	B	Single	Исходное данное
N	N	Integer	Исходное данное
M	M	Integer	Исходное данное
i	i	Integer	Вспомогательная переменная
массив C	C	Single	Результат
k	k	Integer	Результат

3. Составляем фрагмент программы

```
……..
k = 0
For i = 1 To N
 If A(i) < 0 Then
   k = k + 1
   C(k) = A(i)
 End If
Next i
For i = 1 To M
 If B(i) < 0 Then
  k = k + 1
  C(k) = B(i)
 End If
Next i
If k = 0 Then
  MsgBox "Массив C не сформирован",, "Результат"
Else
' Вывод массива C в ячейки.
  Cells(5, 1) = " массив C"
  For i = 1 To k
   Cells(6, i) = C(i)
  Next i
End If
……
```

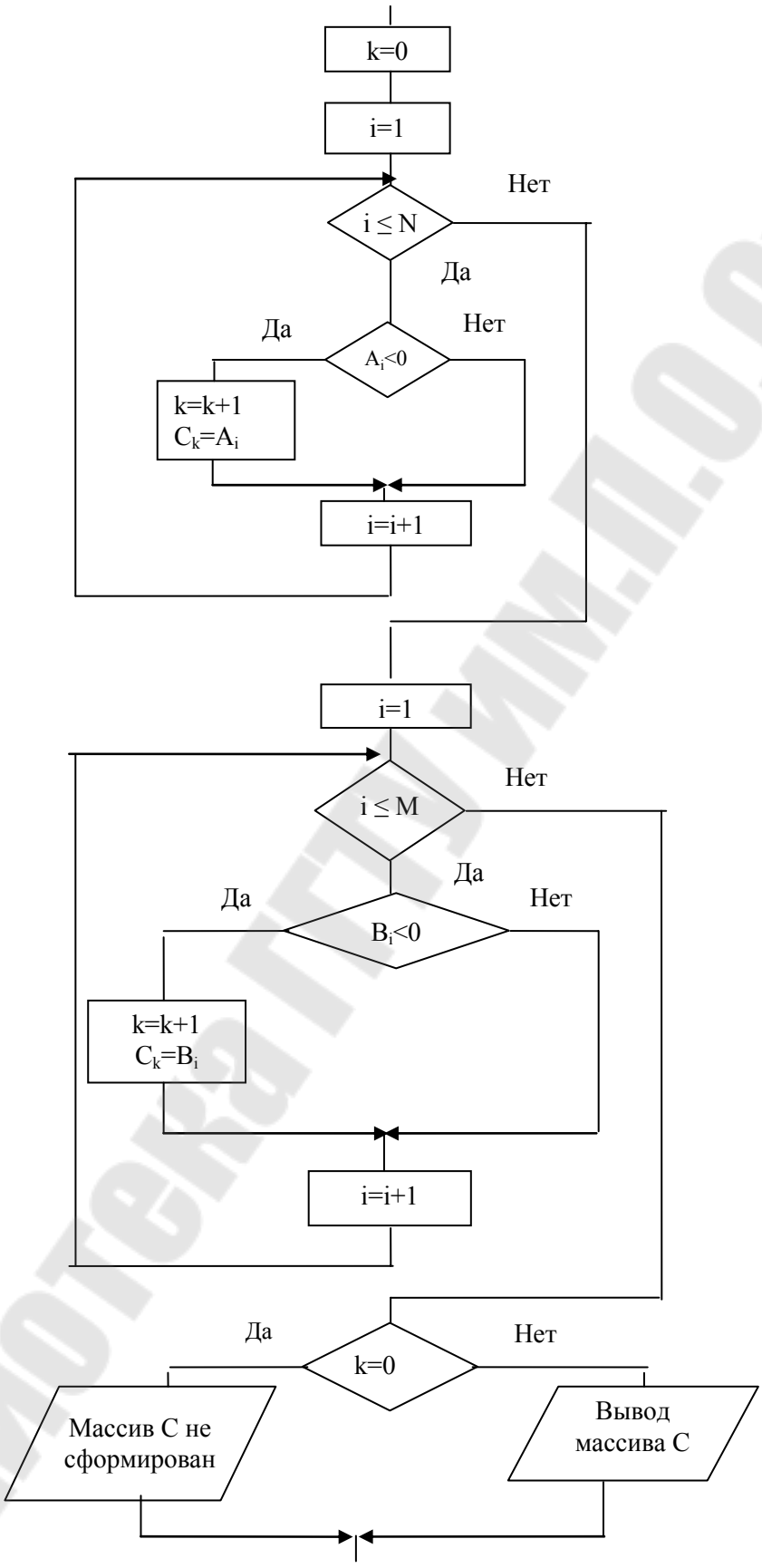


Рис. 3.7. Схема алгоритма примера 3.6

4. Составляем тесты для отладки программы.

Тест 1

Все элементы обоих массивов входят в новый массив.

N=3 M=2

Таблица 3.15

A_1	A_2	A_3
-5	-3	-2

Таблица 3.16

B_1	B_2
-7	-9

Результат: массив С из пяти элементов

Таблица 3.17

C_1	C_2	C_3	C_4	C_5
-5	-3	-2	-7	-9

Тест 2

Нет элементов, входящих в новый массив.

Таблица 3.18

A_1	A_2	A_3
5	3	2

Таблица 3.19

B_1	B_2
7	9

Тест 3

Из обоих массивов в новый массив входит часть элементов.

Таблица 3.20

A_1	A_2	A_3
-5	3	2

Таблица 3.21

B_1	B_2
-4	-2

Результат: массив С из 3-х элементов

Таблица 3.22

C_1	C_2	C_3
-5	-4	-2

Литература

1. Кузьменко, В. Г. VBA 2002 / В. П. Кузьменко. – Москва : БИНОМ, 2002. – 624 с.

2. Водополова, Н. В. Основы алгоритмизации : практ. пособие к лаб. и контрол. работам по курсам «Информатика» и «Основы информатики и вычислительной техники» для студентов всех специальностей днев. и заоч. отделений / авт.-сост. : Н. В. Водополова, В. И. Мисюткин, С. А. Чабуркина. – Гомель : ГГТУ им. П. О. Сухого, 2005. – 32 с.

3. Водополова, Н. В. Обработка одномерных и двухмерных 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.

4. Кравченко, О. А. Основные приемы работы в Excel : практ. пособие для студентов всех специальностей днев. и заоч. отделений / авт.-сост. : О. А. Кравченко, В. И. Мисюткин. – Гомель : ГГТУ им. П. О. Сухого, 2004. – 43 с.

5. Водополова, Н. В. Основы информатики в вычислительной технике. Информатика : практикум к лаб. работам по одноим. курсам для студентов экон. специальностей днев. и заоч. форм обучения / авт.-сост. : Н. В. Водополова, В. И. Мисюткин, С. А. Чабуркина. – Гомель : ГГТУ им. П. О. Сухого, 2005. – 33 с.

Содержание

1. Правила оформления контрольной работы 3
2. Теоретические сведения. Язык Visual Basic for Applications 4
3. Примеры составления программ 15
3.1. Линейный алгоритм 15
3.2. Разветвляющийся алгоритм 17
3.3. Циклический алгоритм табулирования функции 19
3.4. Циклический алгоритм вычисления суммы и произведения элементов массива 22
3.5. Циклический алгоритм нахождения максимального элемента массива и его номера 25
3.6. Циклический алгоритм формирования нового массива из элементов имеющихся 28
Литература 32

Учебное электронное издание комбинированного распространения

Учебное издание

Мисюткин Виктор Иванович

ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ VBA

Методические указания к контрольным работам по курсам «Информатика», «Основы информатики и вычислительной техники» для студентов экономических специальностей заочной формы обучения

Электронный аналог печатного издания

Редактор *Н. И. Жукова*
Компьютерная верстка *Н. Б. Козловская*

Подписано в печать 17.11.08.
Формат 60x84/16. Бумага офсетная. Гарнитура «Таймс».
Ризография. Усл. печ. л. 2,09. Уч.-изд. л. 1,60.
Изд. № 47.
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.