

Обработка одномерных массивов

Для объявления массива на VBA необходимо указать:

- 1) тип элементов массива;
- 2) имя массива;
- 3) его размерность, т.е. число индексов, необходимое для обозначения конкретного элемента.

Каждый массив может разбиваться на несколько подмассивов, которые так же можно разбивать и т.д. Количество таких разбиений, или измерений, называется размерностью массива. Это понятие необходимо отличать от понятия размера массива, определяемого как количество его элементов. При объявлении массива в каждом измерении указывается количество элементов этого измерения.

Объявление одномерного массива имеет вид

Dim <имя_массива>(<количество_элементов>) As <тип_элементов_массива>

Dim x(10) As Single – массив из 10 вещественных чисел x(0) – 1-й элемент, x(1) – 2-й элемент, ..., x(9) – 10-й элемент.

При объявлении массива выделяется участок оперативной памяти, где будут располагаться элементы массива. В оперативной памяти элементы массива размещаются последовательно, друг за другом:

-15	98	1	0,56
a(0)	a(1)	a(2)	a(3)

Каждый элемент массива определяется именем массива и своим порядковым номером, который называется индексом и записывается после имени массива в круглых скобках. Количество индексов, указанных после имени массива при обращении к элементам массива, должно быть равно количеству измерений в массиве. При объявлении массива выделяется участок оперативной памяти для размещения его элементов. Имя массива автоматически содержит адрес начала участка памяти, выделенной для размещения элементов массива, т.е. имя массива – это точка отсчета с точки зрения адресов элементов в памяти. За точку отсчета всегда принимают число 0, поэтому индексу первого элемента массива во внутреннем представлении соответствует число 0, индексу второго – число 1 и т.д. Индекс элемента массива определяет смещение (приращение) адреса заданного элемента массива относительно адреса нулевого элемента массива. Таким образом, нумерация элементов массива всегда начинается с нуля, т.е. индексы элементов изменяются от 0 до N-1, где N – количество элементов в данном измерении.

Шаблон программы, работающей с одномерным массивом можно записать следующим образом:

объявление массива
заполнение массива (ввод или формирование по
заданному закону его элементов)
обработка массива

вывод результата

вывод массива

В языке VBA операции над массивами не определены. Применить какую-либо библиотечную функцию для ввода или вывода всего массива нет возможности. Чтобы ввести или вывести массив, нужно вводить или выводить отдельно сначала первый элемент массива, затем второй, третий и т.д. Таким образом, получается, что ввод-вывод массива – процесс ввода-вывода отдельного элемента, который повторяется несколько раз и зависит от некоторой величины, которая является индексом элемента массива. Следовательно, для ввода-вывода массива нужно использовать цикл с параметром `for...next`. Из выше сказанного следует, что решение задач с использованием массивов предполагает использование циклических структур.

Пример 1. Дан одномерный массив. Найти минимальный и максимальный элементы массива, их порядковые номера и вывести их на печать.

Ход выполнения работы

1. Для нахождения минимального элемента применим стандартный ход. На первом шаге примем за минимальный нулевой элемент массива и номер этого элемента равен нулю. Затем, меняя индексы элементов массива, будем сравнивать найденное минимальное значение с текущим элементом. Если окажется, что текущий элемент меньше минимального значения, найденного на предыдущих шагах, то он становится минимальным и т.д. Для нахождения максимального элемента используют аналогичную схему. Следовательно, для того чтобы проверить каждый элемент массива, нужно использовать цикл с параметром. В алгоритме и программе переменные **min** и **imin** обозначают соответственно минимальное значение и его номер, а переменные **max** и **imax** – максимальное значение и его номер.

2. Написать программу, соответствующую алгоритму:

Алгоритм	Программа
объявление вещ: a(10), min, max; цел: i, imin, imax для i=0 до 10-1 ввод a(i) все для i ‘задаются начальные максимальное и минимальное значения элементов массива и их номера min=a(0), imin=0 max=a(0), imax=0 для i=1 до 10-1 если min>a(i) min=a(i) imin=i все_если если max<a(i) max=a(i)	Private Sub CommandButton1_Click() Dim a(10), min, max As Single, i, imin, imax As Integer For i = 0 To 9 a(i) = Cells(i + 2, 1).Value Next i min = a(0) imin = 0 max = a(0) imax = 0 For i = 1 To 9 If min > a(i) Then min = a(i) imin = i End If If max < a(i) Then max = a(i) End If Next i Print min, imin, max, imax End Sub

```

        imax=i
    все_ если
все_ для i
    печать imin, min
    печать imax, max
    для i=0 до 10-1
        вывод a(i)
    все_ для i

```

```

        imax = i
    End If
Next i
Cells(1, 2).Value = "imin"
Cells(1, 3).Value = "min"
Cells(1, 4).Value = "imax"
Cells(1, 5).Value = "max"

Cells(2, 2).Value = imin
Cells(2, 3).Value = min
Cells(2, 4).Value = imax
Cells(2, 5).Value = max
End Sub

```

Пример 2. Сформировать массив по закону $a_i = \sqrt{i} + \sin i$. При заданном k подсчитать среднее арифметическое элементов массива, больших k .

Ход выполнения работы

1. Средним арифметическим называется отношение суммы чисел к их количеству. Таким образом, для решения задачи нужно найти сумму и количество элементов, удовлетворяющих условию задачи. Так как нужно проверить каждый элемент массива, то необходимо использовать цикл с параметром, где параметр будет обозначать индекс элемента массива. В алгоритме и программе переменные sum и count будут обозначать сумму и количество определенных элементов соответственно.

2. Написать программу, соответствующую алгоритму:

Алгоритм

```

объявление вещ: a(15), sum, k; цел: i, count
ввод k
‘ формирование элементов массива
для i=0 до 15-1
     $a(i) = \sqrt{i} + \sin i$ 
все для i
‘ сначала сумма элементов и их
‘ количество равны нулю
sum=0, count=0
для i=0 до 15-1
    ‘ в цикле каждый элемент
    ‘ сравнивается со значением k
    ‘ нужный элемент добавляется к сумме
    если  $a(i) > k$ 
        sum=sum+a(i)
        count=count+1
    все_ если
все_ для i
‘ находим среднее арифметическое
sum=sum/count
‘ печатаем полученное значение
печать sum
‘ вывод массива на экран
для i=0 до 15-1

```

Программа

```

Private Sub CommandButton2_Click()
Dim a(15), sum, k As Single, i, count As Integer
k = CSng(InputBox("K="))
For i = 0 To 14
    a(i) = Sqr(i) + Sin(i)
Next i
sum = 0
count = 0
For i = 0 To 14
    If a(i) > k Then
        sum = sum + a(i)
        count = count + 1
    End If
Next i
sum = sum / count
MsgBox "Среднее арифметическое элементов, больших k " & CStr(k) & " = " & CStr(sum)
For i = 0 To 14
    Cells(12 + i, 1).Value = a(i)
Next i
End Sub

```

Вывод a(i)
все_для i

```
Private Sub CommandButton2_Click()
Dim a(15), sum, k As Single, i, count As Integer
Dim s As String
s = ""
k = CSng(InputBox("K="))
For i = 0 To 14
    a(i) = Sqr(i) + Sin(i)
    s = s & CStr(a(i)) & Chr(10)
Next i
sum = 0
count = 0
For i = 0 To 14
    If a(i) > k Then
        sum = sum + a(i)
        count = count + 1
    End If
Next i
sum = sum / count
MsgBox " Среднее арифметическое элементов, больших k " & CStr(k) & " = " & CStr(sum) & Chr(10) & s
For i = 0 To 14
    Cells(12 + i, 1).Value = a(i)
Next i

End Sub
```

Пример 3. Дан одномерный массив. Вычислить произведение нечетных элементов массива, расположенных на четных позициях.

Ход выполнения работы

1. Первоначально значение произведения равно 1 (при умножении на 1 значение произведения не меняется). Затем организуется цикл для проверки каждого элемента на соответствие указанному в задаче условию. Если текущий элемент удовлетворяет условию, то с его помощью изменяется искомое произведение.

2. Написать программу, соответствующую алгоритму:

Алгоритм
объявление цел: a(14), цел: i, p
‘ ввод массива с клавиатуры
для i=0 до 14-1
 ввод a(i)
все_для i
 ‘ вычисление произведения
 p=1
 для i=0 до 14-1
 если a(i) mod 2=1 и i mod 2=0
 p=p*a(i)
все_если

Программа
Private Sub CommandButton3_Click()
Dim a(14), p, i As Integer
Dim s As String
s = ""
For i = 0 To 13
 a(i) = CInt(InputBox("a="))
 s = s & CStr(a(i)) & Chr(10)
Next i
p = 1
For i = 0 To 13
 If a(i) Mod 2 = 1 And i Mod 2 = 0 Then

```

все_для i
‘ печатаем полученное значение
печать p
‘ вывод массива a на экран
для i=0 до 14-1
    вывод a(i)
все_для i

```

```

p = p * a(i)
End If
Next i
MsgBox "Произведение элементов = " &
CStr(p) & Chr(10) & s
End Sub

```

Пример 4. Дан одномерный массив $a(15)$. Сформировать новый массив b из элементов массива a , меньших заданного k .

Ход выполнения работы

1. При решении данной задачи нужно знать, сколько элементов должен содержать массив b . Так как массив a вводится с клавиатуры, то значения его элементов заранее неизвестно. В общем случае все элементы массива могут быть меньше заданного k . Предполагаем, что массив b будет содержать столько же элементов, что и массив a . Однако при формировании нового массива не все элемента старого могут удовлетворять условию задачи. По этой причине индексация элементов этих массивов может различаться и, следовательно, для этой цели объявляют разные переменные.

2. Написать программу, соответствующую алгоритму:

Алгоритм

```

объявление вещ: a(15), b(15), k, цел: i, j
ввод k
для i=0 до 15-1
    ввод a(i)
все_для i
j=0 ‘индекс в массиве b
для i=0 до 15-1
    если a(i)<k
        b(j)=a(i)
        j=j+1
все_если
все_для i
для i=0 до 15-1
    вывод a(i)
все_для i
для i=0 до j-1
    вывод b(i)
все_для i

```

Программа

```

Dim a(15), b(15), k As Single, i, j, n As
Integer
Dim sa, sb As String
sa = ""
sb = ""
n = 8
k = CSng(InputBox("K="))
For i = 0 To n - 1
    a(i) = CInt(InputBox("a="))
    sa = sa & CStr(a(i)) & Chr(10)
Next i
j = 0
For i = 0 To n - 1
    If a(i) < k Then
        b(j) = a(i)
        sb = sb & CStr(b(j)) & Chr(10)
        j = j + 1
    End If
Next i
MsgBox "k=" & CStr(k) & Chr(10) &
"Массив a:" & Chr(10) & sa & "Массив b:" &
Chr(10) & sb
End Sub

```

Алгоритмы сортировки одномерных массивов

Под сортировкой понимается упорядочение элементов некоторой последовательности в нужном порядке (убывания или возрастания).

Существует достаточно много алгоритмов сортировок последовательностей. Но мы остановимся только на трёх, наиболее простых.

Метод пузырька (метод обменной сортировки с выбором)

Идея метода отражена в его названии. Шаг сортировки состоит в проходе снизу вверх по массиву. По пути просматриваются пары соседних элементов. Если элементы некоторой пары находятся в неправильном порядке, то они меняются местами. При этом самые «легкие» (наименьшие) элементы массива «всплывают» наверх, а самые «тяжелые» – «тонут». Алгоритмически это можно реализовать следующим образом. Весь массив просматривается снизу вверх, стоящие рядом элементы меняются в том случае, если «нижний» элемент меньше, чем «верхний». Таким образом, наверх «всплывет» самый «легкий» элемент всего массива. Так нужно повторять для оставшихся неотсортированными N-1 элементов (т.е. для тех, которые лежат «ниже» первого) и т.д. Алгоритм достаточно прост:

```
Алгоритм (в порядке возрастания)
объявление вещ: t(10), x, цел: i, j, k, flag
для i=0 до 10-1
    ввод t(i)
все_для i
    для i=10-1 до 1 шаг -1
        'обмена не было
        flag=0
        для j=0 до i-1
            если t(j)>t(j+1)
                'меняем местами два соседних элемента
                x=t(j)
                t(j)=t(j+1)
                t(j+1)=x
                'обмен состоялся
                flag=1
        все_если
    все_для j
    если flag=0
        выход из цикла ' не было обмена
    все_если
все_для i
для i=0 до 10-1
    вывод t(i)
все_для i
```

```
Программа
Private Sub CommandButton5_Click()
Dim t(15), x As Single, i, j, k, n As Integer
Dim flag As Boolean
n = 15
For i = 0 To n - 1
    t(i) = Cells(12 + i, 1).Value
Next i
For i = n - 1 To 1 Step -1
    flag = False
    For j = 0 To i - 1
        If t(j) > t(j + 1) Then
            x = t(j)
            t(j) = t(j + 1)
            t(j + 1) = x
            flag = True
        End If
    Next j
    If Not (flag) Then
        GoTo m
    End If
Next i
m: For i = 0 To n - 1
    Cells(12 + i, 2).Value = t(i)
Next i
End Sub
```

Сортировка выбором

При сортировке этим методом при просмотре массива ищется наименьший элемент, сравнивая его с первым. Если такой элемент найден, но не меняется местами с первым. Затем эти действия повторяются, но не с первого элемента, а со второго. Так продолжается до тех пор, пока не будет отсортирован весь массив:

```
Алгоритм (в порядке возрастания)
объявление вещ: t(10), x, цел: i, j, k
для i=0 до 10-1
```

```
Программа
Private Sub CommandButton6_Click()
Dim t(15), x As Single, i, j, k, n As Integer
```

```

    ввод t(i)
    все_для i
    для i=0 до 10-1
        k=i
        x=t(i)
        для j=i+1 до 10-1
            если t(j)<x
                'меняем местами два элемента
                x=t(j)
                k=j
        все_если
        все_для j
        t(k)=t(i)
        t(i)=x
    все_для i
    для i=0 до 10-1
        вывод t(i)
    все_для i

```

```

n = 15
For i = 0 To n - 1
    t(i) = Cells(12 + i, 1).Value
Next i
For i = 0 To n - 1
    k = i
    x = t(i)
    For j = i + 1 To n - 1
        If t(j) < x Then
            x = t(j)
            k = j
        End If
    Next j
    t(k) = t(i)
    t(i) = x
Next i
For i = 0 To n - 1
    Cells(12 + i, 3).Value = t(i)
Next i
End Sub

```

Двухмерные массивы

Объявление двухмерного массива заключается в последовательной записи типа элементов массива, имени массива и двух измерений. При этом в круглых скобках, соответствующих каждому измерению, указывается количество элементов в данном измерении:

Dim B(5, 10) As Single

В данном примере каждый элемент массива будет вещественным числом, В – имя массива, первое число в круглых скобках указывает количество строк, а второе число – количество столбцов массива. С точки зрения математических определений, массив В – матрица, состоящая из пяти строк и десяти столбцов.

Для обращения к конкретному элементу двухмерного массива необходимо записать имя массива и в круглых скобках последовательно указать индексы элемента массива.

В оперативной памяти элементы массива располагаются так, что при переходе от элемента к элементу наиболее быстро меняется самый правый индекс массива. Если объявлен массив из m строк и n столбцов, то его элементы располагаются в оперативной памяти так, как показано на рис.7.

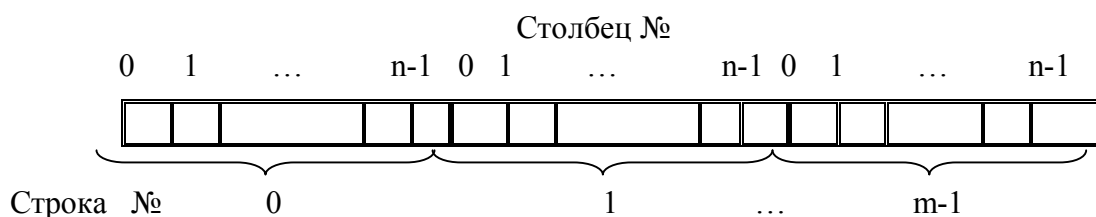


Рис.7

При работе с двумерными массивами следует учесть, что при переходе от элемента к элементу нужно будет менять номер строки и/или номер столбца. В отличие от работы с одномерными массивами решение задач с двумерными массивами требует организации двух циклов: один для изменения номера столбца, другой для изменения номера строки.

Опишем в виде алгоритма и программы отдельно ввод двумерного массива с клавиатуры и его вывод на экран:

Алгоритм

```
объявление вещ: a(4,5), цел: i, j
‘ ввод массива построчно
для i=0 до 4-1
    ‘ в цикле изменяется номер i строки массива
    для j=0 до 5-1
        ‘ в цикле изменяется номер j столбца массива
        ‘ затем элемент вводится
        ввод a(i,j)
    все_для j
все_для i
...
‘ обработка массива (решение задачи)
...
‘ вывод массива построчно
для i=0 до 4-1
    ‘ в цикле изменяется номер i строки массива
    для j=0 до 5-1
        ‘ в цикле изменяется номер j столбца массива
        ‘ затем элемент выводится на экран
        вывод a(i,j)
    все_для j
    ‘ перевод на новую строку
все_для i
```

Среди задач, связанных с использованием двумерных массивов, особое место занимают задачи на формирование массивов, содержащих одинаковое количество строк и столбцов. Такие массивы можно сравнить с квадратными матрицами. Расположение элементов в матрицах определяется, как правило, относительно главной и побочной диагоналей (элементы главной диагонали располагаются на линии, идущей из верхнего левого в правый нижний угол матрицы, а элементы побочной диагонали – на линии, идущей из верхнего правого в левый нижний угол матрицы). Определить такое расположение можно следующим образом.

Элемент расположен:

- на главной диагонали, если $i=j$;
- ниже главной диагонали, если $i>j$;
- выше главной диагонали, если $i<j$;
- на побочной диагонали, если $i=n-j-1$;
- ниже побочной диагонали, если $i>n-j-1$;

➤ выше побочной диагонали, если $i < n - j - 1$,
где i – номер строки, j – номер столбца в матрице размера $n \times n$ (n – количество строк (столбцов) в матрице).

Закон, по которому формируется каждый элемент массива либо задается, либо его необходимо выявить из заданного образца матрицы.

Пример 1. Получить квадратную матрицу порядка n по заданному образцу:

$$\begin{pmatrix} 1 & 2 & \dots & n-1 & n \\ n+1 & n+2 & \dots & 2n-1 & 2n \\ \dots & \dots & \dots & \dots & \dots \\ (n-1)n+1 & (n-1)n+2 & \dots & n^2-1 & n^2 \end{pmatrix}$$

Ход выполнения работы

1. При решении таких задач необходимо выявить закон получения каждого элемента массива. Как правило, значение элемента зависит от номера строки и/или номера столбца. В данной задаче можно увидеть, что в нулевой строке элементы равны номеру столбца плюс 1. С увеличением номера строки на 1 элемент массива увеличивается на число n , а с увеличением номера столбца элемент увеличивается на 1. Из этого сделаем предположение, что элементы массива формируются по закону:

$$a_{ij} = i \cdot n + j + 1$$

2. Написать программу, соответствующую алгоритму:

Алгоритм	Программа
объявление цел: a(4,4), i, j, n	Dim a(4, 4), i, j, n As Integer
ввод n	Dim str As String
' формирование массива построчно	n = CInt(InputBox("Ввод n", "n="))
для i=0 до 4-1	' формирование массива построчно
для j=0 до 4-1	For i = 0 To 4 - 1
a(i,j)=i*n+j+1	For j = 0 To 4 - 1
все_для j	a(i, j) = i * n + j + 1
все_для i	Next j
' вывод массива построчно	Next i
для i=0 до 4-1	' вывод массива построчно
для j=0 до 4-1	For i = 0 To 4 - 1
вывод a(i,j)	For j = 0 To 4 - 1
все_для j	str = str & CStr(a(i, j)) & " "
все_для i	Next j
	str = str & Chr(10)
	Next i
	t = MsgBox(str, , "Вывод массива")

Пример 2. Получить квадратную матрицу порядка n по заданному образцу:

$$\begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 & 1 \\ 0 & 1 & 0 & \dots & 0 & 1 & 1 \\ 0 & 0 & 1 & \dots & 1 & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 1 & \dots & 1 & 1 & 1 \\ 0 & 1 & 0 & \dots & 0 & 1 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 & 1 \end{pmatrix}$$

Ход выполнения работы

1. В этой задаче элементами массива являются 0 и 1. Значения элементов зависят от их месторасположения. Если элемент находится на главной, побочной диагонали или одновременно выше главной и ниже побочной диагонали, то его значение равно 1. В противном случае элемент массива принимает значение 0.

2. Написать программу, соответствующую алгоритму:

Алгоритм

```

объявление цел: a(7,7), цел: i, j
' формирование массива построчно
для i=0 до 7-1
  для j=0 до 7-1
    если i=j или i=n-j-1 или
      (i<j и i>n-j-1) то
      a(i,j)=1
    иначе
      a(i,j)=0
  все_если
все_для j
все_для i
' вывод массива построчно
для i=0 до 7-1
  для j=0 до 7-1
    вывод a(i,j)
  все_для j
все_для i

```

Программа

```

Dim a(7, 7), i, j, n As Integer
Dim str As String
n = CInt(InputBox("Ввод n", "n="))
If n >= 7 Then n = 7
' формирование массива построчно
For i = 0 To n - 1
  For j = 0 To n - 1
    If i = j Or i = n - j - 1 Or (i < j And i > n - j - 1) Then
      a(i, j) = 1
    Else
      a(i, j) = 0
    End If
  Next j
Next i
' вывод массива построчно
For i = 0 To n - 1
  For j = 0 To n - 1
    str = str & CStr(a(i, j)) & " "
  Next j
  str = str & Chr(10)
Next i
t = MsgBox(str, , "Вывод массива")

```

Пример 3. Дана квадратная матрица размера $n \times n$. Вычислить произведение элементов, кратных 3 и расположенных выше главной и ниже побочной диагоналей.

Ход выполнения работы

1. Алгоритм решения данной задачи аналогичен алгоритму предыдущей. Если в примере 2 элемент формируется при выполнении

некоторого условия, то в данной задаче элемент массива, удовлетворяющий заданному условию, участвует в формировании произведения.

2. Написать программу, соответствующую алгоритму:

Алгоритм	Программа
объявление вещ: a(5,5), p, цел: i, j	Dim a(5, 5), i, j, n, p As Integer
' ввод массива построчно	Dim str As String
для i=0 до 5-1	n = 5
для j=0 до 5-1	' ввод массива построчно
ввод a(i,j)	For i = 0 To n - 1
все_для j	For j = 0 To n - 1
все_для i	a(i, j) = CInt(InputBox("Ввод массива"))
' вычисление произведения	Next j
p = 1	Next i
для i=0 до 5-1	' вычисление произведения
для j=0 до 5-1	p = 1
если i < j и i < n-j-1 и остаток от дел	For i = 0 To n - 1
a(i,j) на 3 = 0 то	For j = 0 To n - 1
p=p*a(i,j)	If i < j And i < n - j - 1 And a(i, j) Mod
все_если	3 = 0 Then
все_для j	p = p * a(i, j)
все_для i	End If
печать p	Next j
' вывод массива построчно	Next i
для i=0 до 5-1	' вывод массива построчно
для j=0 до 5-1	For i = 0 To n - 1
вывод a(i,j)	For j = 0 To n - 1
все_для j	str = str & CStr(a(i, j)) & " "
все_для i	Next j
	str = str & Chr(10)
	Next i
	t = MsgBox(str, , "Вывод массива")

Пример 4. Дана матрица размера $n \times n$. Найти минимальный элемент, расположенный выше главной диагонали, поменять его местами с элементом, расположенным в верхнем левом углу матрицы.

Ход выполнения работы

1. Алгоритм решения данной задачи аналогичен алгоритму предыдущей. Надо пройтись по всем элементам матрицы, расположенным выше главной диагонали, и найти среди них наименьший и запомнить его позиции. Затем осуществить обмен соответствующих элементов.

2. Написать программу, соответствующую алгоритму:

Алгоритм	Программа
объявление вещ: a(5,5), p, min; цел: i, j, imin, jmin	Dim a(5, 5), i, j, n, p, imin, jmin, min As Integer
' ввод массива построчно	Dim str As String
для i=0 до 5-1 шаг 1	n = 5
для j=0 до 5-1 шаг 1	For i = 0 To n - 1
ввод a(i,j)	For j = 0 To n - 1
все_для j	a(i, j) = CInt(InputBox("Ввод

```

все_для i
' нахождение минимального элемента и
' его позиции
min=a(0,0), imin=0, jmin=0
для i=0 до 5-1 шаг 1
    для j=0 до 5-1 шаг 1
        если i<j и min>a(i,j)
            min=a(i,j)
            imin=i
            jmin=j
    все_если
все_для j
все_для i
'меняются местами элементы
p =a(0,0)
a(0,0)= a(imin,jmin)
a(imin,jmin)=p
' вывод массива построчно
для i=0 до 5-1 шаг 1
    для j=0 до 5-1 шаг 1
        вывод a(i,j)
    все_для j
все_для i

```

```

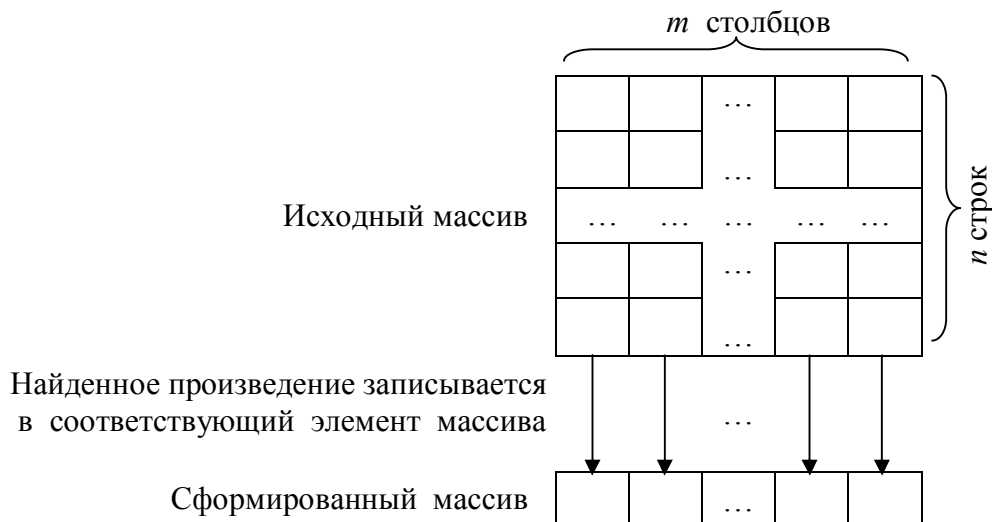
массива"))
Next j
Next i
' нахождение минимального элемента и
' его позиции
min = a(0, 0)
imin = 0
jmin = 0
For i = 0 To n - 1
    For j = 0 To n - 1
        If i < j And min > a(i, j) Then
            min = a(i, j)
            imin = i
            jmin = j
        End If
    Next j
Next i
'меняются местами элементы
p = a(0, 0)
a(0, 0) = a(imin, jmin)
a(imin, jmin) = p
' вывод массива построчно
For i = 0 To n - 1
    For j = 0 To n - 1
        str = str & CStr(a(i, j)) & " "
    Next j
    str = str & Chr(10)
Next i
t = MsgBox(str, , "Вывод массива")

```

Пример 5. Дан двумерный массив размера $n \times m$. Найти произведения нечетных элементов каждого столбца двумерного массива и записать их в одномерный массив. Указать номер столбца с наименьшим произведением элементов.

Ход выполнения работы

1. Решение задачи можно представить в виде схемы:



Как видим, алгоритм будет состоять из трёх основных этапов:

- 1) находим произведение нечетных элементов j -го столбца исходного двумерного массива;
- 2) записываем найденное произведение в соответствующий j -й элемент одномерного массива (сформированный массив будет содержать ровно столько элементов, сколько столбцов в исходном двумерном массиве);
- 3) поиск минимального элемента и его номера в сформированном одномерном массиве.

2. Написать программу, соответствующую алгоритму:

Алгоритм	Программа
объявление вещ: $a(6,4), p, b(4)$, цел: i, j , $pmin, n, m$ $n = 6$ $m = 4$ для $i=0$ до $6-1$ для $j=0$ до $4-1$ ввод $a(i,j)$ все_для j все_для i ‘ вычисление произведений для $j=0$ до $4-1$ ‘ находим произведение j -го столбца $p=1$ для $i=0$ до $6-1$ если остаток от дел $a(i,j)$ на $2 = 1$ $p=p*a(i,j)$ все_если все_для i $b(j)=p$ все_для j $p=b(0)$ $pmin=0$ для $i=1$ до $4-1$ шаг 1 если $p < b(i)$ $p=b(i)$ $pmin=i$ все_если все_для i для $i=0$ до $6-1$ шаг 1 для $j=0$ до $4-1$ шаг 1 вывод $a(i,j)$ все_для j все_для i ‘ вывод массива b для $i=0$ до $4-1$ шаг 1 вывод $b(i)$ все_для i печать $pmin$	$\text{Dim } a(6, 4), p, b(4) \text{ As Single}$ $\text{Dim } i, j, n, m, pmin \text{ As Integer}$ Dim str As String $\text{str} = \text{"Вывод массива A"} \& \text{Chr}(10)$ $n = 6$ $m = 4$ For $i = 0$ To $n - 1$ For $j = 0$ To $m - 1$ $a(i, j) = \text{CInt}(\text{InputBox}(\text{"Ввод массива"}))$ Next j Next i ‘ вычисление произведений For $j = 0$ To $m - 1$ ‘ находим произведение j -го столбца $p = 1$ For $i = 0$ To $n - 1$ If $a(i, j) \text{ Mod } 2 = 1$ Then $p = p * a(i, j)$ End If Next i $b(j) = p$ Next j $p = b(0)$ $pmin = 0$ For $i = 0$ To $m - 1$ If $p < b(i)$ Then $p = b(i)$ $pmin = i$ End If Next i For $i = 0$ To $n - 1$ For $j = 0$ To $m - 1$ $\text{str} = \text{str} \& \text{CStr}(a(i, j)) \& " "$ Next j $\text{str} = \text{str} \& \text{Chr}(10)$ Next i $\text{str} = \text{Chr}(10) \& \text{str} \& \text{" Вывод массива B"} \& \text{Chr}(10)$ For $i = 0$ To $m - 1$ $\text{str} = \text{str} \& \text{CStr}(b(i)) \& " "$

```

Next i
str = str & " Вывод № min" & Chr(10)
str = str & CStr(pmin)
t = MsgBox(str, , "Вывод")

```

Алгоритмы матричной алгебры

Пример 1. Даны две матрицы a и b . Вычислить сумму $c=a+b$, где c – новая матрица.

Ход выполнения работы

1. Для решения данной задачи используется формула $c_{ij} = a_{ij} + b_{ij}$, $i = \overline{1, n}$, $j = \overline{1, m}$ для формирования элементов массива. Каждой матрице в алгоритме соответствует двухмерный массив.

2. Написать программу, соответствующую алгоритму:

```

Алгоритм
объявление вещ: a(6,4), b(6,4), c(6,4), цел:
i, j
ввод массива a построчно
ввод массива b построчно
‘ вычисление суммы
для i=0 до 6-1
    для j=0 до 4-1
        c(i,j)=a(i,j)+b(i,j)
    все_для j
все_для i
вывод массива a построчно
вывод массива b построчно
вывод массива c построчно

```

Пример 2. Дана матрица a и столбец b . Вычислить произведение $c=a*b$, где c – новый столбец.

Ход выполнения работы

1. Для решения задачи используется формула $c_i = a_{ij} \cdot b_j$, $i = \overline{1, n}$, $j = \overline{1, m}$ для формирования элементов массива. Матрице a в алгоритме соответствует двухмерный массив, столбцам b и c – одномерные массивы. Количество элементов массива b должно совпадать с количеством столбцов массива a , а количество элементов массива c – с количеством строк массива a .

2. Написать программу, соответствующую алгоритму:

```

Алгоритм
объявление вещ: a(6,4), b(4), c(6), цел: i, j
ввод массива a построчно
ввод массива b
‘ вычисление произведения
для i=0 до 6-1
    ‘ находим i-ый элемент массива c
    c(i)=0
    для j=0 до 4-1
        ‘ находим элемент массива c
        c(i)=c(i)+a(i,j)*b(j)

```

```

        все_для j
    все_для i
    вывод массива a построчно
    вывод массива b
    вывод массива c

```

Пример 3. Даны матрицы a и b . Вычислить произведение $c=a*b$, где c – новая матрица.

Ход выполнения работы

1. Для решения данной задачи используется формула $c_{ij} = a_{ik} \cdot b_{kj}$, $i = \overline{1, n}$, $j = \overline{1, m}$, $k = \overline{1, p}$ для формирования элементов массива. Матрицам a , b и c в алгоритме соответствуют двумерные массивы. Количество строк массива b должно совпадать с количеством столбцов массива a , а массива c будет иметь размерность $n \times m$, где n – количество строк массива a , m – количество столбцов массива a . Значение p определяет количество столбцов и количество строк массивов a и b соответственно.

2. Написать программу, соответствующую алгоритму:

```

        Алгоритм
    объявление вещ: a(3,4), b(4,5), c(3,5), цел:
    i, j, k
    ввод массива a построчно
    ввод массива b построчно
    ' вычисление произведения
    для i=0 до 3-1
        для j=0 до 5-1
            c(i,j)=0
            для k=0 до 4-1
                c(i,j)=c(i,j)+a(i,k)*b(k,j)
            все_для k
        все_для j
    все_для i
    вывод массива a построчно
    вывод массива b построчно
    вывод массива c построчно

```

Динамические массивы

Память компьютера. Адресное пространство

Память компьютера разбивается на две части: оперативная и внешняя. Внешняя память предназначена для долговременного хранения информации на каком-либо носителе: жестком диске, дискете, компакт-дисках и т.п. Внешняя память компьютера энергонезависима, т.е. при отключении компьютера от электропитания информация с внешних носителей не исчезает. Оперативная память (ОП), наоборот энергозависима, т.е. для сохранения информации требуется постоянное электропитание. Вся ОП разбивается на ячейки одинакового размера, в которых хранится информация. Каждой ячейке поставлено в соответствие конкретное число,

называемое адресом ячейки ОП. Особая роль ОП объясняется тем, что процессор может выполнять команды программы только в том случае, когда они загружены в ОП. Кроме того, в ОП загружаются не только инструкции программы, но и ее данные (переменные). Для распознавания переменных и команд приложения на разных этапах его жизненного цикла используются символьные имена, виртуальные и физические адреса. **Символьные имена** переменных задает программист на этапе написания программы. **Виртуальные адреса** называют логическими, так как они определяются на этапе трансляции программы в машинный код. Поскольку на этом этапе неизвестно, в каком участке ОП будет размещена программа, то переменным и командам программы транслятор присваивает условные (логические) адреса, считая при этом начальный адрес программы нулевым адресом. **Физические адреса** – номера (конкретные адреса) ячеек ОП, в которых будут располагаться команды и переменные программы во время ее работы.

Совокупность всех виртуальных адресов приложения называется виртуальным пространством. При загрузке приложения в ОП виртуальные адреса преобразуются в физические. Процесс преобразования виртуальных адресов в физические заключается в запоминании адреса ячейки ОП, начиная с которой в ОП загружается вся программа для выполнения. Далее адреса команд и переменных, которые установил транслятор, будут представлять собой пару чисел (seg, off), где seg – адрес ячейки, с которой начинает располагаться загруженная программа (сегмент), а off – смещение внутри сегмента (логический адрес, полученный командой или переменной на этапе трансляции).

Участок ОП, выделенный для выполнения программы, называется **адресным пространством** этой программы. Таким образом, каждое приложение работает в своем адресном пространстве.

Динамическая память

При решении задач из предыдущих разделов рассматривались массивы, память под которые отводилась на этапе компиляции программы и сохранялась в течение всей работы программы. Такие массивы называют **статическими**, а выделенную таким образом ОП – **статической**.

Существует другой способ выделения памяти под массивы. Память под переменные можно выделять в процессе выполнения программы. Данный способ выделения памяти называют **динамическим**, переменные, под которые ОП выделялась динамически, – **динамическими**, а выделенная таким способом память – **динамической памятью или кучей**. Динамически распределяемая память (куча) – это свободная область ОП, которая не используется программой при ее загрузке в ОП, операционной системой и другими приложениями. Использование динамической памяти дает дополнительные, по сравнению со статической памятью, возможности: 1) при подключении динамической памяти можно увеличить объем обрабатываемых данных; 2) размер динамических данных можно изменять (или вообще динамические переменные удалить в случае их ненужности).

В то же время динамические переменные не требуют описания в программе, так как память под них выделяется динамически, т.е. в процессе выполнения программы, что делается при помощи указателей.

Динамические массивы

Ранее при определении массивов было сказано, что они представляют собой совокупность элементов, каждый из которых имеет одни и те же атрибуты. Таким образом, все характеристики массива полностью определялись при его объявлении и не могли меняться в течение выполнения программы, поэтому такие массивы будем называть статическими.

При объявлении статического массива:

Dim <имя_массива> (<количество_элементов>) As <тип_элементов_массива>
<имя_массива> содержит нулевой байт области памяти, выделяемой для размещения элементов массива, т.е. <имя_массива> будет содержать адрес первого элемента массива. <количество_элементов> – константное выражение, а <тип_элементов_массива> четко определяет размеры памяти, выделяемой для каждого элемента массива.

Таким образом, количество элементов массива и размеры памяти, выделяемой для него, однозначно задаются при объявлении массива. Однако при решении многих задач необходимо, чтобы память для массива выделялась в процессе выполнения программы, т.е. потребности в памяти заранее не известны и не могут быть определены при объявлении массива.

Массив называется динамическим, если количество его элементов заранее не известно или может изменяться в процессе выполнения программы. При работе с динамическими массивами обязательно выделение и освобождение выделенной динамической памяти. Динамической памятью (кучей) называется участок ОП, где размещаются данные, необходимые для работы программы.

Схему решения задач с использованием динамических массивов можно описать следующим образом.

1. Динамический массив объявляется без указания количества элементов.

2. При получении значения n количества элементов в ходе выполнения программы нужно «переобъявить» массив при помощи оператора ReDim A(n) и ReDim Preserve A(n), если нужно сохранить все элементы, полученные до «переобъявления».

Пример 1. Дан одномерный динамический массив. Вычислить среднее арифметическое модулей элементов массива.

Ход выполнения работы

1. Алгоритмы решения задач с использованием динамических массивов схожи с алгоритмами, использующими статические массивы.

2. Написать программу, соответствующую алгоритму:

Алгоритм	Программа
объявление вещ: a(), sum ; цел: i, n	Dim a(), sum, i, n As Integer
‘ ввод количества элементов массива	Dim str As String
ввод n	str = "Вывод массива A" & Chr(10)
выделить динамическую память под a	n = CInt(InputBox("n="))

```

ввод массива a
sum=0
для i=0 до n-1
    sum=sum+ai
все_для i
    ' находим среднее арифметическое
sum=sum/n
печать sum
вывод массива a

```

```

ReDim a(n)
For i = 0 To n - 1
    a(i) = CInt(InputBox("Ввод массива"))
Next i
sum = 0
For i = 0 To n - 1
    sum = sum + a(i)
Next i
sum = sum / n
For i = 0 To n - 1
    str = str & CStr(a(i)) & " "
Next i
str = Chr(10) & str & "Вывод asred" & Chr(10)
str = str & CStr(sum)
t = MsgBox(str, , "Вывод")

```

Пример 2. Сформировать двухмерный динамический массив по закону $a_{ij} = i + j$.

Ход выполнения работы

1. Алгоритмы решения задач с использованием динамических массивов схожи с алгоритмами, использующими статические массивы.

2. Написать программу, соответствующую алгоритму:

Алгоритм

```

объявление вещ: a(,) ; цел: i, n, m
' ввод количества строк массива
ввод n
' ввод количества столбцов массива
ввод m
выделить динамическую память под a
' формирование массива построчно
для i=0 до n-1
    для j=0 до m-1
        aij=i+j
    все_для j
все_для i
вывод массива a построчно

```

Программа

```

Dim a(), i, j, n, m As Integer
Dim str As String
n = CInt(InputBox("n="))
m = CInt(InputBox("m="))
ReDim a(n, m)
For i = 0 To n - 1
    For j = 0 To m - 1
        a(i, j) = i * n + j + 1
    Next j
Next i
For i = 0 To n - 1
    For j = 0 To m - 1
        str = str & CStr(a(i, j)) & " "
    Next j
    str = str & Chr(10)
Next i
t = MsgBox(str, , "Вывод массива")

```

Пример 3. Дан двухмерный динамический массив размера $n \times m$. Сформировать одномерный динамический массив, каждый элемент которого – произведение нечетных элементов соответствующего столбца двухмерного массива. Указать номер столбца с наименьшим произведением элементов.

Ход выполнения работы

1. Решение этой задачи было подробно рассмотрено в разделе «Двухмерные массивы».

2. Написать программу, соответствующую алгоритму:

Алгоритм	Программа
объявление вещ: a(,),p,b(), цел: i, j, pmin, n, m	Dim a(), p, b() As Single Dim i, j, n, m, pmin As Integer
‘ ввод количества строк массива	Dim str As String
ввод n	str = "Вывод массива A" & Chr(10)
‘ ввод количества столбцов массива	n = CInt(InputBox("n="))
ввод m	m = CInt(InputBox("m="))
выделить динамическую память под a	ReDim a(n, m)
выделить динамическую память под b	ReDim b(m)
‘ ввод массива построчно	For i = 0 To n - 1
для i=0 до 6-1 шаг 1	For j = 0 To m - 1
‘ в цикле изменяется номер i строки	a(i, j) = CInt(InputBox("Ввод массива"))
‘ массива	Next j
для j=0 до 4-1 шаг 1	Next i
ввод a _{ij}	‘ вычисление произведений
все_для j	For j = 0 To m - 1
все_для i	‘находим произведение j-го столбца
‘ вычисление произведений	p = 1
для j=0 до 4-1 шаг 1	For i = 0 To n - 1
‘находим произведение j-го столбца	If a(i, j) Mod 2 = 1 Then
p=1	p = p * a(i, j)
для i=0 до 6-1 шаг 1	End If
‘ находим нечетный элемент	Next i
если остаток от дел a _{ij} на 2 =1	b(j) = p
‘изменяем произведение	Next j
p=p*a _{ij}	p = b(0)
все_если	pmin = 0
все_для j	For i = 0 To m - 1
b(j)=p	If p < b(i) Then
все_для i	p = b(i)
‘находим минимальный элемент	pmin = i
‘массива b	End If
p=b ₀	Next i
‘номер минимального элемента	For i = 0 To n - 1
pmin=0	For j = 0 To m - 1
для i=1 до 4-1 шаг 1	str = str & CStr(a(i, j)) & " "
если p<b _i	Next j
p=b _i	str = str & Chr(10)
pmin=i	Next i
все_если	str = Chr(10) & str & " Вывод массива B"
все_для i	& Chr(10)
‘ вывод массива a построчно	For i = 0 To m - 1
для i=0 до 6-1 шаг 1	str = str & CStr(b(i)) & " "
для j=0 до 4-1 шаг 1	Next i
вывод a _{ij}	str = str & " Вывод № min" & Chr(10)
все_для j	str = str & CStr(pmin)
все_для i	t = MsgBox(str, , "Вывод")
‘ вывод массива b	
для i=0 до 4-1 шаг 1	

вывод b_i
все_для i
печать rmin

Задание 1. Разработать алгоритм и по нему составить два варианта программы для обработки одномерного массива. Первый вариант программы – работа со статическим массивом, второй – с динамическим.

1. Дан массив натуральных чисел $A[N]$. Найти сумму элементов, кратных заданному натуральному K .
2. Дан массив целых чисел $A[N]$, среди которых есть нулевые элементы. Создать массив из номеров таких элементов.
3. Дан массив вещественных чисел $A[N]$, среди которых есть отрицательные элементы. Создать массив из положительных элементов.
4. Дана последовательность целых чисел a_1, a_2, \dots, a_n . Выяснить, какое число встречается раньше — положительное или отрицательное.
5. Дана последовательность действительных чисел a_1, a_2, \dots, a_n . Выяснить, будет ли она возрастающей.
6. Дана последовательность натуральных чисел a_1, a_2, \dots, a_n . Создать массив из четных чисел этой последовательности. Если таких чисел нет, то вывести сообщение об этом факте.
7. Дана последовательность чисел a_1, a_2, \dots, a_n . Указать наименьшую длину числовой оси, содержащую все эти числа.
8. Дана последовательность действительных чисел a_1, a_2, \dots, a_n . Заменить все ее члены, большие данного Z , этим числом. Подсчитать количество замен.
9. Последовательность действительных чисел оканчивается нулем. Найти количество членов этой последовательности.
10. Дан массив действительных чисел, размерность которого N . Подсчитать, сколько в нем отрицательных, положительных и нулевых элементов.
11. Даны действительные числа a_1, a_2, \dots, a_n . Поменять местами наибольший и наименьший элементы.
12. Даны целые числа a_1, a_2, \dots, a_n . Вывести на печать только те числа, для которых $a_i \geq i$.
13. Даны натуральные числа a_1, a_2, \dots, a_n . Указать те из них, у которых остаток от деления на M равен L ($0 \leq L \leq M - 1$).
14. В заданном одномерном массиве поменять местами соседние элементы, стоящие на четных местах, с элементами, стоящими на нечетных местах.
15. При поступлении в вуз абитуриенты, получившие двойку на первом экзамене, ко второму не допускаются. В массиве $A[n]$ записаны оценки экзаменующихся, полученные на первом экзамене. Подсчитать, сколько человек не допущено ко второму экзамену.
16. Дана последовательность чисел, среди которых имеется один нуль. Вывести на печать все числа до нуля включительно.

17. В одномерном массиве размещены: в первых элементах — значения аргумента, в следующих — соответствующие им значения функции. Напечатать элементы этого массива в виде двух параллельных столбцов (аргумент и значения функции).
18. Пригодность детали оценивается по размеру B , который должен соответствовать интервалу $(A - \delta, A + \delta)$. Определить, имеются ли в партии из N деталей бракованные. Если да, то подсчитать их количество, в противном случае выдать отрицательный ответ.
19. У вас есть доллары. Вы хотите обменять их на рубли. Есть информация о стоимости купли-продажи в банках города. В городе N банков. Составьте программу, определяющую, какой банк выбрать, чтобы выгодно обменять доллары на рубли.
20. Дан целочисленный массив с количеством элементов n . Напечатать те его элементы, индексы которых являются степенями двойки (1, 2, 4, 8, 16, ...).
21. Задана последовательность из N вещественных чисел. Определить, сколько среди них чисел меньших K , равных K и больших K .
22. Задана последовательность N вещественных чисел. $S_i = \sqrt{\frac{(X_i - M)^2}{N - 1}}$. Вычислить где M — среднее арифметическое данной последовательности.
23. Задан массив действительных чисел. Определить, сколько раз меняется знак в данной последовательности чисел, запомнить номера позиций, в которых происходит смена знака.
24. Задана последовательность N вещественных чисел. Вычислить сумму чисел, порядковые номера которых являются четными числами.
25. Задана последовательность N вещественных чисел. Вычислить произведение чисел, порядковые номера которых являются нечетными числами.

Задание 2. Найти закон, по которому формируется указанная матрица. Разработать алгоритм и по нему составить программу для формирования квадратной матрицы порядка n . Первый вариант программы — работа со статическим массивом, второй — с динамическим.

$$1. \begin{pmatrix} 1 & 2 & 3 & \dots & n \\ n & n-1 & n-2 & \dots & 1 \\ 1 & 2 & 3 & \dots & n \\ n & n-1 & n-2 & \dots & 1 \\ \dots & \dots & \dots & \dots & \dots \\ n & n-1 & n-2 & \dots & 1 \end{pmatrix}$$

$$2. \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 & 1 \\ 0 & 0 & 0 & \dots & 0 & 2 & 0 \\ 0 & 0 & 0 & \dots & 3 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & n-1 & 0 & \dots & 0 & 0 & 0 \\ n & 0 & 0 & \dots & 0 & 0 & 0 \end{pmatrix}$$

$$3. \begin{pmatrix} n & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & n-1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & n-2 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & 2 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{pmatrix}$$

$$4. \begin{pmatrix} 1 \cdot 2 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 2 \cdot 3 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 3 \cdot 4 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & (n-1)n & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & n(n+1) \end{pmatrix}$$

$$5. \begin{pmatrix} 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 0 & 0 & \dots & 0 & 0 & 1 \\ 1 & 1 & 1 & \dots & 1 & 1 & 1 \end{pmatrix}$$

$$6. \begin{pmatrix} 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ 2 & 2 & 2 & \dots & 2 & 2 & 0 \\ 3 & 3 & 3 & \dots & 3 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ n-1 & n-1 & 0 & \dots & 0 & 0 & 0 \\ n & 0 & 0 & \dots & 0 & 0 & 0 \end{pmatrix}$$

$$7. \begin{pmatrix} 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ 0 & 1 & 1 & \dots & 1 & 1 & 0 \\ 0 & 0 & 1 & \dots & 1 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 1 & 1 & \dots & 1 & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 & 1 \end{pmatrix}$$

$$8. \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 & 1 \\ 1 & 1 & 0 & \dots & 0 & 1 & 1 \\ 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 0 & \dots & 0 & 1 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 & 1 \end{pmatrix}$$

$$9. \begin{pmatrix} n & 0 & 0 & \dots & 0 & 0 & 0 \\ n-1 & n & 0 & \dots & 0 & 0 & 0 \\ n-2 & n-1 & n & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 2 & 3 & 4 & \dots & n-1 & n & 0 \\ 1 & 2 & 3 & \dots & n-2 & n-1 & n \end{pmatrix}$$

$$10. \begin{pmatrix} 1 & 2 & 3 & \dots & n-2 & n-1 & n \\ 2 & 3 & 4 & \dots & n-1 & n & 0 \\ 3 & 4 & 5 & \dots & n & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ n-1 & n & 0 & \dots & 0 & 0 & 0 \\ n & 0 & 0 & \dots & 0 & 0 & 0 \end{pmatrix}$$

$$11. \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 & n \\ 0 & 2 & 0 & \dots & 0 & n-1 & 0 \\ 0 & 0 & 3 & \dots & n-2 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 2 & 0 & \dots & 0 & n-1 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 & n \end{pmatrix}$$

12.

$$\begin{pmatrix} 1 & 2 & 3 & \dots & n-2 & n-1 & n \\ 2 & 1 & 2 & \dots & n-2 & n-2 & n-1 \\ 3 & 2 & 1 & \dots & n-4 & n-3 & n-2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ n-1 & n-2 & n-3 & \dots & 2 & 1 & 2 \\ n & n-1 & n-2 & \dots & 3 & 2 & 1 \end{pmatrix}$$

13. Дано действительное число x .

$$\begin{pmatrix} 1 & x & x^2 & \dots & x^{n-3} & x^{n-2} & x^{n-1} \\ x & 0 & 0 & \dots & 0 & 0 & x^{n-1} \\ x^2 & 0 & 0 & \dots & 0 & 0 & x^{n-2} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x^{n-2} & 0 & 0 & \dots & 0 & 0 & x \\ x^{n-1} & x^{n-2} & x^{n-3} & \dots & x^2 & x & 1 \end{pmatrix}$$

14. Даны действительные числа a_1, a_2, \dots, a_n .

$$\begin{pmatrix} a_1 & a_2 & a_3 & \dots & a_{n-2} & a_{n-1} & a_n \\ a_2 & a_3 & a_4 & \dots & a_{n-1} & a_n & a_1 \\ a_3 & a_4 & a_5 & \dots & a_n & a_1 & a_2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n-1} & a_n & a_1 & \dots & a_{n-4} & a_{n-3} & a_{n-2} \\ a_n & a_1 & a_2 & \dots & a_{n-5} & a_{n-4} & a_{n-3} \end{pmatrix}$$

$$15. \begin{pmatrix} 1 & 2 & 3 & \dots & 9 & n \\ 0 & 1 & 2 & \dots & 8 & n-1 \\ 0 & 0 & 1 & \dots & 7 & n-2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix}$$

$$16. \begin{pmatrix} 1 & 0 & \dots & 0 & 1 \\ 0 & 1 & \dots & 1 & 0 \\ \dots & . & \dots & . & \dots \\ 0 & 1 & \dots & 1 & 0 \\ 1 & 0 & \dots & 0 & 1 \end{pmatrix}$$

$$17. \begin{pmatrix} 0 & 1 & 2 & \dots & n-1 \\ 0 & 0 & 1 & 2 & \dots & n-2 \\ 0 & 0 & 0 & 1 & 2 & \dots & n-3 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix}$$

$$18. \text{ Дан одномерный массив } x_1, x_2, \dots, x_n. \begin{pmatrix} x_1 & x_2 & \dots & x_{n-1} & x_n \\ x_1^2 & x_2^2 & \dots & x_{n-1}^2 & x_n^2 \\ . & . & \dots & . & . \\ x_1^n & x_2^n & \dots & x_{n-1}^n & x_n^n \end{pmatrix}$$

19. Дан одномерный массив x_1, x_2, \dots, x_n .

$$\begin{pmatrix} 1 & 1 & \dots & 1 & 1 \\ x_1 & x_2 & \dots & x_{n-2} & x_n \\ x_1^2 & x_2^2 & \dots & x_{n-1}^2 & x_n^2 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ x_1^{n-1} & x_2^{n-1} & \dots & x_{n-1}^{n-1} & x_n^{n-1} \end{pmatrix}$$

$$20. \begin{pmatrix} 1 & 2 & \dots & n-1 & n \\ n+1 & n+2 & \dots & 2n-1 & 2n \\ 2n+1 & 2n+2 & \dots & 3n-1 & 3n \\ \dots & \dots & \dots & \dots & \dots \\ (n-1)n+1 & (n-1)n+2 & \dots & n^2-1 & n^2 \end{pmatrix}$$

$$21. \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 2 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & n-1 \end{pmatrix}$$

$$22. \begin{pmatrix} n & n-1 & n-2 & \dots & 1 \\ n-1 & n-2 & n-3 & \dots & 0 \\ n-2 & n-3 & n-4 & \dots & -1 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 0 & -1 & \dots & 2-n \end{pmatrix}$$

$$23. \begin{pmatrix} 0 & 2 & 4 & \dots & 2n-2 \\ 2 & 4 & 6 & \dots & 2n \\ 4 & 6 & 8 & \dots & 2n+2 \\ \dots & \dots & \dots & \dots & \dots \\ 2n-2 & 2n & 2n+2 & \dots & 4n-4 \end{pmatrix}$$

$$24. \begin{pmatrix} 25 & -1 & -2 & \dots & 1-n \\ 0 & 25 & -1 & \dots & -n \\ 0 & 0 & 25 & \dots & -n+1 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 25 \end{pmatrix}$$

Задание 3. Разработать алгоритм и по нему составить два варианта программы для обработки двумерного массива. Первый вариант программы – работа со статическим массивом, второй – с динамическим.

1. Вычислить сумму и число положительных элементов матрицы $A[N][N]$, находящихся над главной диагональю.
2. Дана матрица A размером $n \times m$. Определить k — количество особых элементов массива A , считая его элемент особым, если он больше суммы остальных элементов его столбца.
3. Задана квадратная матрица. Поменять местами строку с максимальным элементом на главной диагонали со строкой с заданным номером k .
4. Дана матрица $B[N][M]$. Найти в каждой строке матрицы максимальный и минимальный элементы и поменять их местами с первым и последним элементом строки соответственно.
5. Дана целая квадратная матрица n -го порядка. Определить, является ли она магическим квадратом, т. е. такой, в которой суммы элементов во всех строках и столбцах одинаковы.
6. Дана матрица размером $n \times m$. Переставляя ее строки и столбцы, добиться того, чтобы наибольший элемент (или один из них) оказался в верхнем левом углу.
7. Определить, является ли заданная целая квадратная матрица n -го порядка симметричной (относительно главной диагонали).
8. Дана целочисленная квадратная матрица. Найти в каждой строке наибольший элемент и поменять его местами с элементом главной диагонали.
9. Упорядочить по возрастанию элементы каждой строки матрицы размером $n \times m$.
10. Задана матрица размером $n \times m$ и целое число $k < n$, $k < m$. Найти максимальный по модулю элемент матрицы. Переставить строки и столбцы матрицы таким образом, чтобы максимальный по модулю элемент был расположен на пересечении k -й строки и k -го столбца.
11. Дана квадратная матрица $A[N][N]$. Записать на место отрицательных элементов матрицы нули, а на место положительных - единицы. Вывести на печать нижнюю треугольную матрицу в общепринятом виде.
12. Дана действительная матрица размером $n \times m$, все элементы которой различны. В каждой строке выбирается элемент с наименьшим значением, затем среди этих чисел выбирается наибольшее. Указать индексы элемента с найденным значением.
13. Дана действительная квадратная матрица порядка N (N — нечетное), все элементы которой различны. Найти наибольший элемент среди стоящих на главной и побочной диагоналях и поменять его местами с элементом, стоящим на пересечении этих диагоналей.
14. Для заданной квадратной матрицы сформировать одномерный массив из ее диагональных элементов. Найти след матрицы, суммируя элементы

одномерного массива. Преобразовать исходную матрицу по правилу: четные строки разделить на полученное значение, нечетные оставить без изменения.

15. Задана квадратная матрица. Получить транспонированную матрицу.

16. Квадратная матрица, симметричная относительно главной диагонали, задана верхним треугольником в виде одномерного массива. Восстановить исходную матрицу и напечатать по строкам.

17. Заданы матрица порядка n и число k . Разделить элементы k -й строки на диагональный элемент, расположенный в этой строке.

18. Для целочисленной квадратной матрицы найти число элементов, кратных k , и наибольший из них.

19. Найти наибольший и наименьший элементы прямоугольной матрицы и поменять их местами.

20. Дана прямоугольная матрица. Найти строку с наибольшей и наименьшей суммой элементов. Вывести на печать найденные строки и суммы их элементов.

21. В данной действительной квадратной матрице порядка n найти сумму элементов строки, в которой расположен элемент с наименьшим значением. Предполагается, что такой элемент единственный.

22. В данной действительной квадратной матрице порядка n найти наибольший по модулю элемент. Получить квадратную матрицу порядка $n-1$ путем отбрасывания в исходной матрице строки столбца, на пересечении которых расположен элемент с найденным значением.

23. Дана действительная квадратная матрица порядка n . Преобразовать матрицу по следующему правилу: строку с номером n сделать столбцом с номером n , а столбец с номером n - строкой с номером n .

24. Пусть дана действительная матрица размером $n \times m$. Требуется преобразовать матрицу следующим образом: поэлементно вычесть последнюю строку из всех строк, кроме последней.