

Работа с одномерными массивами (подсчет, поиск элементов, перестановки в массиве)

Задание (общее ко всем вариантам)

Составить программу обработки одномерного массива заданного типа произвольной длины. Количество используемых данных должно вводиться с клавиатуры (в пределах отведенного под массив места). Значения элементов массива вводятся или с клавиатуры, или из предварительно созданного текстового файла (в соответствии с заданием). Исходный массив и результаты его обработки выводить в выводной текстовый файл.

Оформить отчет по работе.

Требования к программе

- Если специально не указано, что данные в массив вводятся с клавиатуры, перед запуском программы необходимо подготовить данные требуемого типа в отдельном текстовом файле, которому следует присвоить имя, совпадающее с именем файла программы, но с расширением ".DAT". Данные набивать, отделяя друг от друга пробелом, по 10 – 20 чисел в строке. Для символьных данных – все набивать в одной строке.
- Программа должна обрабатывать данные указанного типа в количестве до 50 чисел и до 250 символов, если количество не указано в задании. Конкретный размер массива при запуске программы вводить с клавиатуры.
- Все значения, на которые по смыслу накладываются ограничения, должны при вводе контролироваться.
- Вывод исходных данных и результатов производить в выходной текстовый файл. При выводе использовать длину выводимой строки не более 76 символов.

Содержание программы

- Заголовок программы с комментарием;
- Описание переменных и массива;
- Описание меток;
- Ввод исходных данных с клавиатуры;
- Открытие входного и выводного файлов;
- Заполнение массива из файла;
- Вывод на печать введенного массива в выводной файл под заголовком "Исходные данные";
- Проведение обработки массива в соответствии с заданием;
- Вывод результатов обработки в выводной файл под заголовком "Результаты расчета";
- Закрытие всех файлов.

Общие пояснения

Рассмотрим некоторые моменты, связанные с обработкой одномерных массивов. Следует помнить, что выделение памяти под массив при его описании может производиться только с помощью констант (в явном виде, или с предварительным описанием константы), например:

```
Const MIN=5;  
      MAX=100;  
Var Massiv1 : array[1..110] of real;  
    Massiv2 : array[MIN..MAX] of integer;
```

Кроме того, рекомендуется сначала завести свои описатели типов массивов, а затем с их помощью выделять место под конкретные переменные этих "массивных" типов, например:

```
Const MAXI=200;  
Type IntMasMAX = array[1..MAX] of integer; {описатель для целочисленных массивов  
из MAX элементов}  
      RealMass = array[1..110] of real; { описатель  
для вещественных массивов из 110 элементов }  
Var Massiv1,Nmbs:IntMasMAX; {завели 2 целочисл. массива}  
    Massiv2,WindV:RealMass; {Завели 2 веществ. массива}
```

Программы, использующие массивы, позволяют сначала ввести все необходимые данные, а только потом их использовать. Поэтому в таких программах следует предусматривать следующие шаги (в указанном порядке):

1. **Ввод исходных данных.** Обычно включает ввод количества элементов массива (с проверкой допустимости введенного значения) и ввод самих элементов массива в указанном количестве. Если ввод значений предусмотрен с клавиатуры, перед каждым оператором чтения должен быть запрос на ввод. Если


```

End;
. . .
Close(Fout); {закрытие файла }

```

3. **Обработка массива.** Здесь могут встретиться различные ситуации, на которые следует обратить внимание.

а) Если в задаче предлагается использовать признак делимости на некоторое число значения переменной (например, требуется что-то делать с каждым элементом массива, который нацело делится на 5), то следует в цикле брать каждый элемент, и проверять на равенство нулю остатка от деления значения элемента массива на 5:

```

for i:=1 to N do
begin
  if K[i] mod 5 = 0 then
    { делать что надо, т.к. элемент массива делится на 5 } ;
. . .
end;

```

б) Если же предлагается использовать признак делимости номера элемента, то разумнее сначала вычислить количество выбираемых номеров и организовать цикл по вычисленному количеству, каждый раз определяя, какой элемент массива надо обрабатывать.

Например, если надо что-то делать с каждым пятым элементом массива K длиной N элементов, то цикл обработки будет иметь вид:

```

M:=N div 5; { определение числа повторений цикла }
for i:=1 to M do
begin
  j:=i*5; { определение номера обрабатываемого элемента }
  { обработка K[j]-го элемента массива }
. . .
end;

```

Если нужно обрабатывать каждый пятый элемент, начиная с третьего, программа будет выглядеть по-другому:

```

j:=3;
M:=(N-(3-1)) div 5; {определение числа повторений цикла}
for i:=1 to M do
begin { обработка K[j]-го элемента массива }
. . .
  j:=j+5; {определение номера следующего обраб.элемента}
end;

```

в) Если требуется обрабатывать символьный массив, например, подсчитать, сколько раз встречается символ '©', нужно уметь, во-первых, присваивать символьным переменным нужные значения, во-вторых, вводить символы в массив с клавиатуры, и, в-третьих, уметь сравнивать символьные элементы массива.

Присваивание значения переменной можно сделать либо перенося это значение из другой переменной (если она содержит в данный момент нужный символ), либо задавая присваивание константы, либо используя функцию `Chr(i)`, преобразования целого числа в символ (с кодом, равным этому числу). Таблица кодов символов приведена в *Приложении Г*. Примеры присваивания:

```

Const
  SimA ='A';
  Paragraf = #21;
  Kod = 21;
  SixKod = $15;
Var S1,S2,S,S4,S5,S6 : char;
. . .

```

```

Begin
. . .
S1:= SimA;
S2:=S1;
S3:='A';
S3:=Paragraf;
S4:=#21;
S5:=#$15;
S6:=Chr(21);
S7:=Chr($15);

```

Пример используемых операторов:

```

. . .
Var Sim : char;

```

```

TextM:array[1..100] of char;{завели символьный массив}
i,N : shortint;
. . .
Begin
. . .
{Если заполнение символьного массива TextM с клавиатуры}
for i:=1 to N do
begin
writeln('Вводи очередной символ');
readln(TextM[i]);
end;
. . .
{Если заполнение символьного массива TextM из файла}
assign(fin, 'UMNIK4.DAT');
reset(fin);
for i:=1 to N do
read(fin,TextM[i]);
. . .
close(fin); { не забыть в конце закрыть все открытые файлы }

```

4. **Вывод результатов.** Если в процессе выполнения программы исходный массив изменяется (в нем меняются сами значения элементов, их количество или они переставляются), исправленный массив должен выводиться в конце программы под заголовком 'Измененный массив'.

Разбор контрольного варианта

Задание.

Таблица 1. Данные к заданию 31 варианта

№ вар.	Задание	Печатать элементы массива по		Тип обработ. данных
		штук:	формату:	
31	Формирование символьного массива длиной не более ста элементов, заполнение его с клавиатуры (вводя по одному произвольному символу пока не встретится символ "."), подсчет и вывод на экран символа, который встретился чаще других и число его повторений.	30	:2	Символьный

При решении этой задачи нужно выделить место на символьный массив из ста элементов, заполнить его (в итеративном цикле – пока не встретится символ '.' или не введется сто символов), распечатать его (по 30 символов в строке) и провести подсчеты частоты вхождения каждого символа. Последняя часть представляет наибольшую сложность, поэтому остановимся на ней подробнее.

Так как нужно найти самый частый символ и количество его повторений, необходимо в программе завести переменные символьного типа для рассматриваемого символа (**SimI**) и для самого частого символа (**SimMax**), а также счетчики для рассматриваемого (**Ni**) и самого частого (**Nmax**) символов. В качестве **SimI** будем брать по очереди каждый символ из массива (во внешнем цикле по **i** – номеру в массиве длиной **N** элементов) и для него считать, сколько раз этот символ встречается в массиве (внутренний цикл по **j** – для всех элементов массива). Если после подсчета **Ni** окажется больше **Nmax**, значение **Ni** переносится в **Nmax**, а **SimI** – в **SimMax**.

Очевидно, перед внешним циклом **Ni** следует обнулить, а в **SimMax** занести, например, первый элемент символьного массива.

Длина массива **N** подсчитывается один раз при заполнении массива (и не может превышать 250).

Понятно, что данный алгоритм не самый эффективный, так как любой символ, встречающийся несколько раз, мы будем обчислять столько раз, сколько он встречается. Но пренебрежение этим обстоятельством позволит существенно упростить алгоритм, так как не нужно запоминать и каждый раз в начале тела внешнего цикла проверять, какие символы мы уже обчислили.

Для разбираемого варианта не приводятся таблица идентификаторов и блок-схема алгоритма, хотя при оформлении отчета их следует включить.

С учетом выполненного выше разбора задачи, программа может иметь вид:

Текст программы

```
PROGRAM Simbol_Array;
VAR
  N,Ni,Nmax,i,j :integer;
  SimI,SimMax   :char;
  Simbol        :array[1..100] of char;
  Fout          :text;
BEGIN {Выполняемый блок. Первый раздел - ввод данных }
  N:=0;
  writeln('Вводить по одному не более 100 символов,');
  writeln(' Для завершения - ввести символ "."');
  repeat
    writeln('Введи очередной символ');
    N := N + 1;
    readln(Simbol[N]);
  until (N = 100) or (Simbol[N] = '.');
  if Simbol[N]= '.' then N := N - 1; { Последний символ, если он - точка,
  рассматриваться не будет }
  if N < 1 then
    writeln('Нельзя рассматривать меньше 1 символа')
  else
    begin
      Assign(Fout,'UMNIK4.RES'); {связали Fout с UMNIK.DAT}
      Rewrite(Fout);             {открыли файл для записи }
      writeln(Fout,'Исходный массив из ',N,' элементов');
      for i:=1 to N do
        begin
          write(Fout,Simbol[i]:2); { печать в текущей строке}
          if i mod 30 = 0 then writeln(Fout); { если номер
          элемента кратен 30 переходим на новую строку}
        end;
      writeln(Fout);
      { Основной раздел - поиск номера }
      Nmax := 0; { количество наиболее частого символа }
      SimMax := Simbol[1];
      for i:=1 to N do { внешний цикл перебора символов }
        begin
          SimI := Simbol[i];
          Ni := 0;
          for j:=i to N do { внутренний цикл перебора символов }
            if Simbol[j] = SimI then Ni:=Ni+1;
            if Ni>Nmax then {если этот символ встретился чаще,}
              begin {запомним его и число его вхождений}
                Nmax := Ni;
                SimMax := SimI;
              end;
            end;
          end;
      { Раздел вывода результатов поиска }
      if Nmax = 1 then
        writeln(Fout,'Все символы входят по 1 разу')
      else
        write(Fout,'Символ "',SimMax,'" встретился ',Nmax,
          ' раз');
        { далее определим, если Nmax кончается на 2,3,4 и не во втором десятке, изменим
        окончание, например, "23 раза" }
        if (Nmax mod 10 < 5) and (Nmax mod 10 > 1) and
          (Nmax div 10 <> 1)
          then writeln(Fout,'a');
        end;
      Close(Fout);
    END.
```

Результаты работы

(содержимое файла UMNIK.RES)

Исходный массив из 95 элементов

Не говорит мне , он умер - он живет ! П
усть жертвенник разбит - огонь еще п
ылает , пусть роза сорвана - она еще ц
ветет

Символ "е" встретился 14 раз

Варианты заданий

Таблица 2. Варианты заданий

№ вар.	Задание	Печатать элементы массива по		Тип обрабатываемых данных
		шпук:	формату:	
1	Формирование массива длиной не более ста элементов, заполнения его с клавиатуры (вводя по одному произвольному символу пока не встретится символ '!'), исключение из массива всех символов – цифр и пробелов со сдвигом остающихся элементов массива на освободившиеся места.	30	:1	Символьн.
2	Нахождение номера наименьшего элемента в массиве заданной длины среди всех положительных элементов. Длина массива вводится с клавиатуры.	5	:11	Веществ.
3	Вычисление суммы "К" слагаемых – элементов массива, начиная с третьего по порядку, и суммируя только элементы с нечетными номерами. Длина массива и количество суммируемых элементов вводится с клавиатуры.	6	:9:2	Веществ.
4	Перестановка максимального и минимального по значению элементов массива. Значение длины массива вводится с клавиатуры.	8	:7	Целые
5	Вычисление суммы всех четных (по значению) элементов массива, расположенных на нечетных по порядку местах. Длина массива вводится с клавиатуры.	10	:6	Целые
6	Нахождение максимального значения в массиве среди всех четных чисел. Длина массива вводится с клавиатуры.	9	:6	Целые
7	Вычисление суммы элементов массива, начиная с "К"-го по порядку и до элемента, равного нулю (если такой элемент встретится; иначе – до конца массива). "К", длину массива и значения элементов вводить с клавиатуры.	5	:12	Веществ.
8	Перестановка "i"-го и "j"-го по порядку элементов массива, при условии, что они с разными знаками. Если они с одинаковыми знаками, все элементы между ними обнулить. Длина массива и номера переставляемых элементов вводятся с клавиатуры.	10	:6	Целые
9	Вычисление суммы элементов массива, расположенных в конце массива, причем складывать нужно, начиная от конца массива, столько элементов, пока сумма не превысит значения 20.5 (или не будут сложены все элементы). Длина массива вводится с клавиатуры. Печатать сумму и количество сложенных элементов	6	:10:3	Веществ.
10	Нахождение максимального значения в массиве среди всех элементов после первого отрицательного. Длина и значения элементов массива вводятся с клавиатуры.	10	:5	Целые
11	Вычисление суммы элементов массива начиная с первого элемента со значением больше 0.9, и пока сумма по модулю не превысит заданного значения. Длина массива, значения элементов и предельное значение (признак для окончания суммирования) вводятся с клавиатуры.	5	:9:3	Веществ.
12	Перестановка максимального и минимального по коду символов массива длиной N элементов. Значение длины массива и сами элементы массива вводятся с клавиатуры. Массив печатать до и после перестановки.	30	:2	Символьн.
13	Формирование массива длиной N элементов, заполняя его с клавиатуры (вводя сначала количество символов, и затем по одному произвольному символу пока не введется указанное количество), подсчет и вывод на экран количества символов из диапазона от 'А' до 'я' (кириллицы).	30	:2	Символьн.

№ вар.	Задание	Печатать элементы массива по		Тип обработыв данных
		шпук:	формату:	
14	Вычисление среднего значения в одномерном массиве для всех элементов между первым и вторым нулем в массиве (или от первого нулевого значения до конца массива). Длина и значения элементов массива вводятся с клавиатуры.	5	:12	Веществ.
15	Перестановка одномерного массива в обратном порядке. Значение длины массива вводится с клавиатуры. Массив печатать до и после перестановки.	9	:8	Целые
16	Вычисление в одномерном массиве целой части среднего значения всех положительных четных по величине чисел. Длина массива вводится с клавиатуры.	8	:8	Целые
17	Исключение из массива всех пробелов и запятых со сдвигом остающихся элементов массива на освободившиеся места. Исходный массив длиной N символов (где N не больше 100) вводится из текстового файла.	30	:1	Символьн.
18	Вычисление целого среднearифметического значения среди всех отрицательных элементов массива (содержащего и положительные значения), расположенных начиная с "K"-го по порядку элемента. Длина массива и значение "K" вводятся с клавиатуры.	10	:6	Целые
19	Определение длины самой длинной последовательности из расположенных подряд в одномерном логическом массиве значений TRUE и вывод найденной длины на экран. Длина массива и значения TRUE и FALSE вводятся с клавиатуры в форме <i>F</i> (для FALSE) и <i>T</i> (TRUE).	8	:7	Логич.
20	Вычисление среднearифметического всех положительных элементов массива (содержащего и отрицательные значения), длина которого вводится с клавиатуры.	6	:10:2	Веществ.
21	Поиск места (номера элемента) в массиве, где первый раз подряд встречаются два четных числа. Длина массива вводится с клавиатуры.	10	:6	Целые
22	Формирование логического массива длиной N элементов, заполнения его с клавиатуры (вводя 1 – вместо TRUE и 0 – вместо FALSE), подсчет и вывод на экран количества значений TRUE и FALSE и сообщение, чего было больше.	10	:6	Логич.
23	Поиск номеров двух последних расположенных подряд отрицательных элементов в массиве. Длина массива вводится с клавиатуры.	6	:10	Веществ.
24	Формирование логического массива, заполняя его с клавиатуры (вводя вместо TRUE четные числа, а вместо FALSE – нечетные и заканчивая, когда встретится число 0); подсчет и вывод на экран количества значений TRUE и FALSE и сообщение, каких значений было больше.	12	:5	Логич.
25	Нахождение номера максимального значения в массиве после первого отрицательного и не далее второго отрицательного. Длина массива вводится с клавиатуры.	5	:10:3	Веществ.
26	Формирование символьного массива длиной N элементов, содержащего только цифры. (При заполнении его с клавиатуры пропускать не цифровые символы.) Подсчет и вывод на экран цифры, встретившейся чаще других (и сколько раз она встретилась.)	25	:3	Символьн.
27	Исключение из массива всех отрицательных по значению элементов со сдвигом элементов массива на освободившиеся места. Длина массива вводится с клавиатуры.	6	:9:2	Веществ.

№ вар.	Задание	Печатать элементы массива по		Тип обработыв данных
		ступок:	формату:	
28	Переписывание из исходного в выходной массив всех нечетных значений в начало, а всех четных – в конец. Длина исходного массива вводится с клавиатуры.	10	:6	Целые
29	Определение номера элемента, с которого начинается самая длинная последовательность четных чисел, расположенных в массиве подряд. Длина массива вводится с клавиатуры	9	:6	Целые
30	Определение номера элемента, с которого начинается самая длинная последовательность из отрицательных чисел, расположенных в массиве подряд. Длина массива вводится с клавиатуры.	6	:10:2	Веществ.
31	Формирование символьного массива длиной не более ста элементов, заполнение его с клавиатуры (вводя по одному произвольному символу пока не встретится символ "."), подсчет и вывод на экран символа, который встретился чаще других и число его повторений	30	:2	Символьн.