

Лабораторная работа № 5

Обработка одномерных массивов

Задание: составить программу заданной обработки массива целых чисел. В процессе обработки использовать перестановки элементов внутри массива, не создавая новых массивов. Заполнение исходного массива организовать с помощью генератора случайных чисел, если иное не предусмотрено вариантом задания. Исходный и обработанный массив выводить на экран.

Теоретический материал

При выполнении лабораторных работ 5 и 6 будет использоваться структурированный тип данных *массив*, который представляет собой фиксированное количество упорядоченных однотипных компонент, снабженных индексами. Он может быть *одномерным* и *многомерным*. В лабораторной работе № 5 будут использоваться одномерные массивы, в лабораторной работе № 6 – двумерные.

Тип-массив описывается в разделе описания типов следующим образом:

type <имя типа> = **array** [<тип индекса (индексов)>] **of** <тип компонент>;

Размерность массива может быть любой, компоненты массива могут быть любого (в том числе и структурированного) типа, индекс (индексы) может (могут) быть только интервального или перечисляемого типа. При употреблении в качестве типа индекса типа *Integer* или *Word* можно использовать только его диапазон.

При описании типа индекса (индексов) можно использовать константы, которые должны быть определены до определения типа.

Определенный в разделе описания типов тип-массив можно использовать для описания переменных и типизированных констант. Тип-массив можно вводить непосредственно и при определении соответствующих переменных или типизированных констант.

При задании значений константе-массиву компоненты указываются в круглых скобках и разделяются запятыми, причем, если массив многомерный, внешние круглые скобки соответствуют левому индексу, вложенные в них круглые скобки – следующему индексу и т. д. При этом все компоненты массива должны быть заполнены.

Доступ к компонентам массива осуществляется указанием имени массива, за которым в квадратных скобках помещается значение индекса (индексов) компоненты. В многомерных массивах значения индексов перечисляются через запятую.

Для работы с массивом как с единым целым используется идентификатор массива без указания индексов в квадратных скобках. Массив может участвовать только в операциях отношения («=», «<>») и в операторе присваивания.

Массивы, участвующие в этих действиях, должны быть идентичны по структуре, то есть иметь одинаковые базовые типы и типы индексов.

Задание значений переменной типа массив может осуществляться с клавиатуры, путем генерирования случайным образом либо иным способом. Когда значения элементов задаются с клавиатуры или генерируются случайным образом, необходима организация цикла, в котором последовательно происходит обращение к каждому элементу массива. Вывод значений элементов массива на экран или в файл также происходит в цикле.

При написании программ необходимо следить за тем, чтобы значения индексов не превышали границ, указанных при объявлении массива, так как выход индекса за границы массива приводит к сбою в работе программы. Контроль значений индексов массивов можно организовать при помощи директивы компилятора `{ $R+ }`, которая приводит к проверке всех индексных выражений на соответствие их значений диапазону индекса.

Пример выполнения лабораторной работы

Для примера рассмотрим следующую задачу. Дан одномерный массив целых чисел. Удалить все отрицательные элементы, расположенные справа от максимального.

Прежде всего, определим алгоритм решения данной задачи. Вначале надо найти максимальный элемент среди элементов массива и запомнить его индекс. Затем, начиная со следующего за максимальным элементом, проверять элементы массива на знак, и если число оказывается отрицательным, то удаляем этот элемент из массива путем «сдвига» элементов массива влево и уменьшаем длину массива. При этом следует заметить, что в памяти переменная типа массив будет занимать столько же места, как и до удаления элементов.

В графическом виде этот алгоритм будет блок-схему, представленную на рис. 5.1.

Теперь пишем код программы.

Поскольку в программе требуется использование генератора случайных чисел, то необходимо подключить стандартный модуль CRT, в котором описана функция генерации случайных чисел.

```
Uses crt;
```

В разделе описания типов определим тип-массив:

```
type arr=array[1..15] of integer;
```

В разделе описания переменных описываем необходимые переменные:

```
var a:arr;                {массив}
    i,j,n: byte;         {индексы, длина массива}
    i_max: byte;        {индекс максимального элемента массива}
    max:integer;       {максимальный элемент}
```

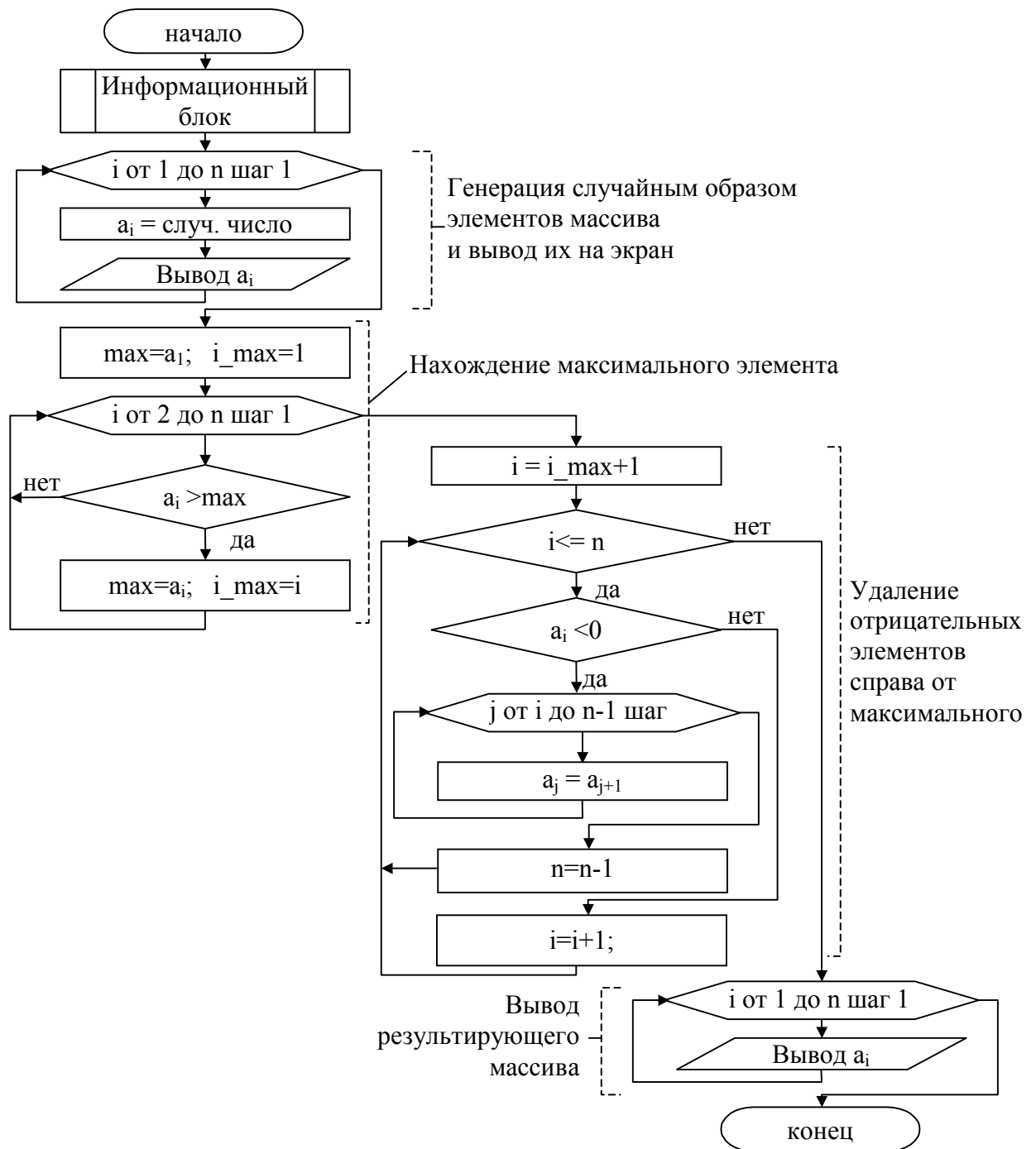


Рис. 5.1. Блок-схема алгоритма удаления из массива отрицательных элементов, расположенных справа от максимального элемента массива

Далее описываем основное тело программы. Как отмечено на рисунке, алгоритм состоит из нескольких частей. Первая часть – генерация исходного массива:

```

begin
  writeln;
  writeln('      Автор – Иванов И.П., студент гр. ИСЭд-11');
  writeln('      Вариант № 100');
  writeln(' Дан одномерный массив целых чисел. ');

```

```

writeln(' Удалить все отрицательные элементы, расположенные
справа от максимального ');
writeln;
{$R+}      {Включаем контроль значений индексов}
clrscr;
n:=15;
randomize;
writeln('Исходный массив:');
for i:=1 to n do      {генерация элементов массива}
begin
  a[i]:=-20+random(41); {в диапазоне [-20; 20]}
  write(a[i]:4);
end;
writeln;

```

Генератор случайных чисел активизируется командой `randomize`. Далее в цикле случайным образом задаются элементы массива. Функция `random(41)` возвращает случайное число в диапазоне от 0 до 40. Получается, что минимально возможное значение элементов массива будет равно -20 (к -20 прибавить 0, сгенерированный функцией `random`), максимально возможное – 20 (-20 плюс число 40, сгенерированное функцией `random`). Сразу же выводим значение элемента массива на экран. В цикле все элементы будут выводиться в строчку, т. к. используется команда вывода `write`. После генерации и вывода всех элементов массива осуществляется переход на новую строку.

Вторая часть алгоритма – поиск максимального элемента. Предположим, что первый элемент – максимальный, тогда заппомним значение первого элемента в переменной `max`, а в переменную `i_max`, где будет запоминаться положение максимального элемента, запишем 1. Далее организуется цикл, в котором ищется элемент, больший чем `max`.

```

max:=a[1];
i_max:=1;
for i:=2 to n do
  if a[i]>max then
  begin
    max:=a[i];
    i_max:=i;
  end;

```

Далее, начиная с элемента, следующим за максимальным, организуется цикл проверки элементов массива на знак (важно заметить, что в данном случае нужно использовать цикл `while`, т. к. длина массива будет изменяться, и, соответственно, число итераций цикла также будет меняться). Если число оказывается отрицательным, то удаляем этот элемент из массива, для чего организуется еще один цикл, в котором на место удаляемого записывается стоящий следом элемент.

Внутренний цикл сдвига можно проиллюстрировать на примере удаления седьмого элемента из массива, содержащего 10 элементов (рис. 5.2).

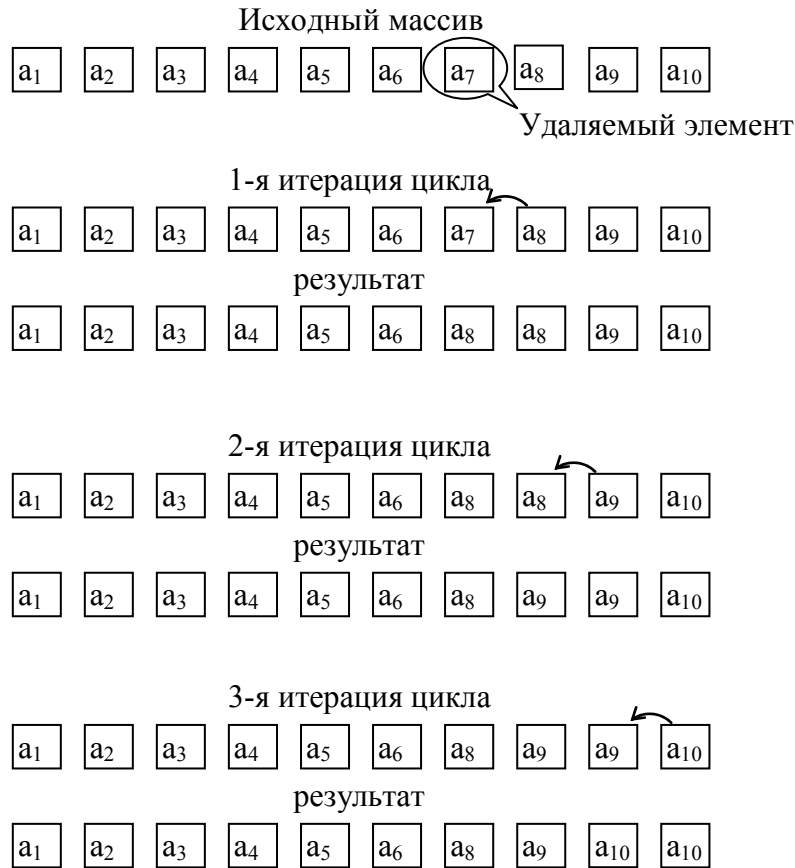


Рис. 5.2. Схема удаления элемента из массива

В программе удаление элементов, находящихся справа от максимального, будет выглядеть следующим образом:

```

i:=i_max;
while i<=n do
begin
  if a[i]<0 then
  begin
    for j:=i to n-1 do
      a[j]:=a[j+1];
    n:=n-1;
  end
  else
    i:=i+1;
end;

```

Еще одна тонкость данного фрагмента состоит в том, что переход к следующему элементу ($i:=i+1$) для проверки осуществляется только тогда, когда элемент не удовлетворяет условию. Если же i -й элемент окажется отрицательным, то

после удаления на его месте окажется $i+1$ -й элемент, который и нужно проверить на следующей итерации цикла.

Остается вывести результирующий массив на экран.

```
writeln('Результат обработки:');
for i:=1 to n do
    write(a[i]:4);
readln;
{$R-}      {Выключаем контроль значений индексов}
end.
```

Ниже представлен полный текст программы.

```
program lab5_var100;
Uses crt;
type arr=array[1..15] of integer;
var a:arr;           {массив}
    i,j,n: byte;    {индексы, длина массива}
    i_max: byte;    {индекс максимального элемента массива}
    max:integer;    {максимальный элемент}
begin
    writeln;
    writeln('      Автор - Иванов И.П., студент гр. ИСЭд-11');
    writeln('      Вариант № 100');
    writeln(' Дан одномерный массив целых чисел. ');
    writeln(' Удалить все отрицательные элементы, расположенные
справа от максимального ');
    writeln;
    {$R+}      {Включение контроля значений индексов}
    clrscr;   {Очистка экрана}
    n:=15;
    randomize;
    writeln('Исходный массив:');
    for i:=1 to n do          {генерация элементов массива}
    begin
        a[i]:=-20+random(41); {в диапазоне [-20; 20]}
        write(a[i]:4);
    end;
    writeln;
    max:=a[1];
    i_max:=1;
    for i:=2 to n do          {поиск максимального элемента массива}
        if a[i]>max then
            begin
                max:=a[i];
                i_max:=i;
            end;
    end;
```

```

    end;
i:=i_max+1;
while i<=n do          {цикл для проверки элементов на знак }
begin
    if a[i]<0 then      {если элемент отрицательный, то}
    begin
        for j:=i to n-1 do {цикл удаления i-го элемента}
            a[j]:=a[j+1];
        n:=n-1;          {уменьшение длины массива}
        end
    else
        i:=i+1;         {иначе переход к следующему элементу}
    end;
writeln('Результат обработки:');
for i:=1 to n do
    write(a[i]:4);
readln;
{$R-}      {Выключение контроля значений индексов}
end.

```

Варианты заданий

1. Скорректировать массив $A=(a_1, a_2, \dots, a_n)$, переписав в начало массива группу, содержащую наибольшее число подряд идущих положительных элементов. Элементы массива вводить с клавиатуры.
2. В массиве $A=(a_1, a_2, \dots, a_n)$ все элементы, равные нулю, поставить сразу после максимального элемента данного массива. Элементы массива вводить с клавиатуры.
3. В массиве $A=(a_1, a_2, \dots, a_n)$ все отрицательные элементы отправить в «хвост» массива.
4. В массиве $A=(a_1, a_2, \dots, a_n)$ удалить последнюю группу положительных элементов. Группой называется подряд идущие элементы одного знака, число которых больше или равно 2.
5. В массиве $A=(a_1, a_2, \dots, a_n)$ все положительные элементы, стоящие перед минимальным положительным элементом, переслать в «хвост» массива.
6. В массиве $A=(a_1, a_2, \dots, a_n)$ удалить все подряд идущие отрицательные элементы, идущие вслед за минимальным элементом массива.
7. В массиве $A=(a_1, a_2, \dots, a_n)$ удалить все отрицательные элементы, стоящие перед минимальным элементом массива.
8. В массиве $A=(a_1, a_2, \dots, a_n)$ удалить все элементы, меньшие, чем элемент массива, расположенный слева от максимального.

9. В массиве $A=(a_1, a_2, \dots, a_n)$ вставить новый элемент со значением P вслед за наибольшим из отрицательных элементов этого массива.
10. В массиве $A=(a_1, a_2, \dots, a_n)$ удалить все элементы, стоящие между минимальным положительным и максимальным отрицательным элементами.
11. В массиве $A=(a_1, a_2, \dots, a_n)$ удалить все положительные элементы, имеющие четный порядковый номер, идущие после минимального элемента массива.
12. В массиве $A=(a_1, a_2, \dots, a_n)$ все положительные элементы, начиная со второго положительного, отправить в «хвост» массива.
13. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ группу, содержащую наибольшее число равных элементов, заменить на максимальный элемент этого массива. После корректировки массив может содержать меньше элементов, чем прежде. Элементы массива вводить с клавиатуры.
14. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ группу элементов, содержащую наибольшее число подряд идущих отрицательных элементов, переписать в «хвост» массива. Элементы массива вводить с клавиатуры.
15. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ все отрицательные элементы, имеющие нечетный порядковый номер, отправить в «хвост» массива, т. е. поместить на место последних элементов.
16. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ все группы элементов, содержащие более 3-х подряд идущих отрицательных элементов, заменить на максимальный элемент. Элементы массива вводить с клавиатуры.
17. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ все положительные элементы, имеющие четный порядковый номер, переписать в начало массива.
18. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ группу, содержащую наибольшее число равных элементов, заменить на максимальный элемент этого массива. Элементы массива вводить с клавиатуры.
19. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ удалить все отрицательные элементы, расположенные между положительными.
20. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ исключить из массива группу с наибольшим числом подряд идущих положительных элементов. Элементы массива вводить с клавиатуры.
21. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ отрицательные элементы, имеющие четный порядковый номер, переписать в начало массива.
22. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ удалить все равные элементы, оставив только один из данных групп равных. Элементы массива вводить с клавиатуры.
23. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ группу из наибольшего числа подряд идущих нулей заменить на максимальный элемент массива. Элементы массива вводить с клавиатуры.

24. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ группу, содержащую наибольшее число подряд идущих положительных элементов, переписать в «хвост» массива.
25. В одномерном массиве $A=(a_1, a_2, \dots, a_n)$ все положительные элементы, расположенные между отрицательными, поставить после минимального элемента массива.