



Школа Программирования

*Богомолова Ольга Борисовна,
Усенков Дмитрий Юрьевич*

ПРАКТИКУМ ПРОГРАММИРОВАНИЯ: ПОИГРАЕМ В «РЕВЕРСИ» НА МАССИВАХ!

Реверс массива (его зеркальное отражение) – операция интересная. В практике программирования она встречается не очень часто, но иногда бывает нужна – например при реализации зеркального отражения растровых изображений. При этом стандартных («библиотечных») функций, реализующих реверс массива, в большинстве языков программирования не предусмотрено, и приходится всё делать руками. А это не так просто и, как показала практика, вызывает трудности у учеников.



Рассмотрим программу (листинг 1), которая выполняет реверс части массива от элемента с индексом, записанным в переменную x , до элемента с индексом, записанным в переменную y .

Алгоритм реверса:

1) рассматриваем фрагмент от x -го до y -го элемента, при этом обязательно x должно быть меньше y , иначе (если интервал реверса был по ошибке задан наоборот) меняем эти переменные местами;

2) если $x = y$, то ничего делать не нужно, поэтому вся ключевая часть алгоритма записана в ветви **then** условного оператора с условием неравенства x и y ;

3) при выполнении реверса работаем в цикле, который выполняется, пока x остается меньше y (если же x станет равен или больше y , то цикл завершается);

4) меняем местами элементы с индексами x (первый в текущем интервале от x до y) и y (последний в этом интервале);

5) «стягиваем» интервал, отбрасывая уже обработанные первый и последний элементы, для чего x увеличиваем на 1, а y , наоборот, уменьшаем на 1;

6) если после этого x всё еще меньше y , то цикл повторяется уже для новых «начального» и «конечного» элемента текущего «остатка» от интервала реверса;

Листинг 1

```

program revers;
var a: array [1..10] of integer;
    // объявляем массив и рабочие переменные
    i,x,y,t: integer;
begin
for i:= 1 to 10 do // вводим массив
    readln(a[i]);

    readln(x); // вводим индекс переменной, от которой начинается реверс
    readln(y); // вводим индекс переменной, до которой производится реверс

    if x>y then begin t:=x; x:=y; y:=t; end; // обеспечиваем, что всегда x<y
                                                // ключевая часть алгоритма реверса

    if x<>y then // выполняем реверс, только если x не равен y
        while (x<y) do begin // цикл выполняется, пока x остается меньше y
            t:=a[x]; // меняем местами элементы с индексами x и y,
            a[x]:=a[y]; // то есть текущие «первый» и «последний»
            a[y]:=t;
            x:=x+1; // сужаем фрагмент, увеличивая x на 1 и уменьшая y на 1
            y:=y-1;
        end; // цикл заканчивается, когда x = y («середина»)
            // или если x > y («проскочили середину»)

    for i:=1 to 10 do // вывод полученного массива
        writeln(a[i]);
end.

```

7) если же после увеличения x и уменьшения y окажется, что новый интервал сжался до одного элемента (среднего), то есть $x = y$, или если мы «проскочили» середину и оказалось, что x стал больше y , то цикл завершается, так как условие в **while** станет ложным – значит, реверс завершен.

Пример задачи

Выполнить реверс отдельно первой и второй половины массива.

Решение

См. листинг 2 (Pascal ABC.Net).

Конечно, удобнее всего алгоритм реверса оформить в виде подпрограммы (процедуры) и вызывать его по мере надобности. Сделать это мы оставляем самим читателям.

Листинг 2

```

program revers;
const N=10; // обязательно четное значение количества элементов
var a: array [1..N] of integer; // здесь объявляем массив и рабочие переменные
    i,k,m,t: integer;
begin
    randomize;
    for i:= 1 to N do // заполняем массив
        a[i] := Round(random*(10-1)+1); // целые случайные числа от 1 до 10

    // вывод исходного массива
    writeln("Исходный массив:");
    for i:=1 to N do // вывод исходного массива
        write(a[i], ' ');
    writeln;

```

```
// реверс первой половины массива:
k := 1;           // индекс переменной, от которой начать реверс
m := N div 2;    // индекс переменной, до которой вести реверс
// алгоритм реверса

if k<>m then      // выполняем реверс, только если k не равен m
  while (k<m) do begin // цикл выполняем, пока k остается меньше m
    t:=a[k];        // меняем местами элементы с индексами k и m,
    a[k]:=a[m];     // то есть текущие «первый» и последний»
    a[m]:=t;
    k:=k+1;         // сужаем фрагмент, увеличивая k на 1 и уменьшая m на 1
    m:=m-1;
  end;              // цикл кончится, если k = m («середина»)
                   // или если k > m («проскочили середину»)

// реверс второй половины массива:
k := (N div 2)+1; // индекс переменной, от которой начать реверс
m := N;           // индекс переменной, до которой вести реверс

// алгоритм реверса

if k<>m then      // выполняем реверс, только если k не равен m
  while (k<m) do begin // цикл выполняем, пока k остается меньше m
    t:=a[k];        // меняем местами элементы с индексами k и m
    a[k]:=a[m];     // то есть текущие «первый» и последний»
    a[m]:=t;
    k:=k+1;         // сужаем фрагмент, увеличивая k на 1 и уменьшая m на 1
    m:=m-1;
  end;              // цикл кончится, если k = m («середина»)
                   // или если k > m («проскочили середину»)

// вывод полученного массива
writeln("Полученный массив:");
for i:=1 to N do   // вывод полученного массива
  write(a[i], ' ');
writeln;
end.
```



*Богомолова Ольга Борисовна,
доктор педагогических наук,
учитель математики
и информатики
ГБОУ СОШ № 1360, г. Москва,*

*Усенков Дмитрий Юрьевич,
Московский государственный
институт индустрии туризма
имени Ю.А. Сенкевича, г. Москва.*