

A decorative frame consisting of two thick black L-shaped lines. One L-shape is in the top-left corner, and the other is in the bottom-right corner, framing the central text.

ВСТАВКА И УДАЛЕНИЕ ЭЛЕМЕНТОВ ОДНОМЕРНОГО МАССИВА

Вставка одного элемента

Пример 1. В массив, состоящий из n элементов, вставить число b после k -го элемента.

- **Решение**

- Рассмотрим на конкретном примере. Пусть дан следующий одномерный массив из 10 ($n = 10$) элементов:

3, -12, 5, 14, 27, -6, 1, -34, 10, -15.

- Вставим число 100 после 5 элемента ($b=100$, $k = 5$)

Алгоритм вставки элемента в массив:

- Первые 5 элементов массива остаются без изменений.
- Сдвинуть все элементы, начиная с шестого, на один назад. Но чтобы не потерять соседнее значение, сдвиг будем начинать с последнего элемента — сдвинуть сначала десятый на один вправо, затем девятый, восьмой и т. д. до шестого ($m[i+1]:=m[i]$, $i=n..k$).
- На место шестого элемента записываем значение 100, то есть после 5-го элемента массива ($m[k+1]:=b$).

Получим следующий массив:

- **3, -12, 5, 14, 27, 100, -6, 1, -34, 10, -15.**

- Таким образом, в массиве стало 11 элементов, то есть массив надо определять на $N+1$ элемент:

- `Array[1..n+1] Of Integer.`

Фрагмент программы:

- **For $i:=n$ Downto $k1+1$ Do {сдвиг элементов на одну позицию назад}**
- **$m[k1+1]:= x1$; {вставка элемента на место после $k1$ -го}**

Пример 2. Вставить число после всех элементов массива, кратных 3.

Решение

1. Необходимо обратить внимание на описание массива: на сколько элементов может увеличиться массив? Максимальное количество элементов, после которых может быть вставлен новый элемент, совпадает с количеством элементов массива.

Так как может случиться, что все элементы массива отвечают заданному свойству. Поэтому массив может увеличиться в два раза (это его самая большая размерность), а значит, соответствующее ему описание будет следующим:

Array[1..2*n] Of Integer;

2. Если мы будем просматривать элементы массива с начала и вставлять новый после элемента с заданным свойством, то следующим просматриваемым элементом будет новый (вставленный) элемент и его необходимо будет пропускать («перепрыгивать»). Поэтому решение будет не очень эффективным.

Поэтому лучше всего просматривать массив, начиная с конца, тогда вставляемый элемент мешать не будет. При этом просмотр будет последовательным от N-го до 1-го.

3. Номер последнего элемента после каждой вставки будет меняться. Его нужно будет переопределять. Для этого нужно вести подсчет количества вставленных элементов на данный момент.

Для вставки нескольких элементов в массив составим программу, в которой будет вестись подсчет количества вставленных элементов и корректироваться номер последнего элемента.

Номер последнего элемента необходим для того, чтобы знать, сколько элементов необходимо сдвинуть при освобождении места для нового элемента, так как количество элементов в этой части массива увеличивается.

- WriteLn(' Введите число для вставки: ');
- ReadLn(x);
- k: = 0;
- For i:= n Downto 1 Do
 - If M[i] Mod 3 = 0 Then
 - {n+k - номер последнего элемента в данный момент}

 - For j:= n+k Downto k1+1 Do {сдвиг элементов на одну позицию назад}
 - m[j+1]:= m[j]; m[k1+1]: = x1; {вставка элемента после k1-го}
 - Inc(k); {счётчик вставленных элементов}

 - {вывод массива после вставки всех элементов}

Удаление элементов из массива

- Задача 1. Необходимо удалить k -ый элемент из массива m , состоящего из n элементов.

Решение.

Пусть дан одномерный массив, состоящий из 10 элементов:

- 9, 6, 7, 10, 14, **16**, 5, 11, 4, 8.
- Нужно удалить элемент с номером 6 ($n=10, k=6$).
Алгоритм удаления шестого элемента заключается в следующем:
- Сместить все элементы, начиная с 6-го, на один элемент влево: 6-му присвоить значение 7-го, 7-му присвоить значение 8-го, 8-му присвоить значение 9-го, 9-му присвоить значение 10-го:
 $m[6]:=m[7]; m[7]:=m[8]; m[8]:=m[9]; m[9]:=m[10]$. На этом сдвиг заканчивается.
Таким образом, сдвиг начинается с k -го элемента и идет по $(n-1)$ -й элемент:
 $m[i]:=m[i+1], i=k..n-1$
- Последнему элементу массива присвоить значение, равное 0: $m[n]:=0$.
- При дальнейшей работе с этим массивом использовать $n-1$ элемент.
- После удаления элемента из массива, изменится количество элементов в массиве (уменьшается на один) и у части элементов изменится индекс.
- После выполнения алгоритма, массив будет следующим:
- **9, 6, 7, 10, 14, 5, 11, 4, 8, 0.**

{ k — номер удаляемого элемента.} {смещение элементов, начиная с k -го, на один влево}
for $i := k$ **to** $n-1$ **do** $m[i] := m[i+1];$
 $m[n]:=0;$ {последнему элементу присваиваем 0}

Задача . Удалить из массива все максимальные элементы.

Решение

В этом случае лучше массив начинать просматривать с конца, так как иначе нужно будет снова возвращаться к элементу с номером, который только что обработали.

Эта может случиться тогда, когда подряд идут два или более максимальных элементов. Если первый удалим, то на его место сдвигается следующий максимальный элемент (индекс изменять не будет необходимости).

Если просматривать массив с конца, то на место удаленного элемента встанет уже обработанный элемент, и можно будет перейти к обработке следующего элемента (путем изменения индекса на один).

Для организации просмотра массива с конца, можно воспользоваться оператором цикла For с убывающей переменной цикла:

For i := n Downto 1 Do <тело цикла>, где значение переменной i уменьшается от n до 1.

Для решения задачи делаем **два просмотра** массива.

Первый просмотр массива делаем с начала для того, чтобы найти значение максимального элемента (на этот раз нужно определить значение элемента, а не его индекс). Для этого можно использовать процедуру Maximum. У неё два выходных параметра: значение максимального элемента и его индекс.

Второй просмотр производится с конца массива для того, чтобы удалить максимальные элементы. Перебираем все элементы, если текущий элемент массива имеет максимальное значение, то удаляем его, значение счётчика удаленных элементов k увеличим на 1.

Подсчитываем количество удаленных элементов для того, чтобы знать, сколько элементов выводить на экран.

- ...

- **k:=0;**

{нахождение значения максимального элемента массива и его номера

- **For i:=n Downto 1 Do If A[i] = max Then** {если текущий элемент равен максимальному, то} **Begin**

- **for j := i to n-1 do**

- m[j] := m[j+1]; m[n]:=0;**

- {удаляем его}

- **Inc(k);** {увеличиваем счетчик удаленных элементов}

- **End;**

- Исходный массив - ~~9~~, 6, 7, ~~9~~, ~~9~~, 5, 1, 1, 4, ~~9~~.

- Итоговый массив – 6, 7, 5, 1, 1, 4, 0, 0, 0, 0