

Слияние 2 упорядоченных массивов

- Имеются два упорядоченных по возрастанию (предыдущий элемент меньше последующего) массива. Требуется получить третий упорядоченный по возрастанию массив, путем слияния первых двух.
- **Например.**
- Если
1-й массив: 4, 8, 12, 14, 23, 85,
а 2-й массив: 2, 4, 8, 9, 12, 16,
тогда 3-й массив будет таким: 2, 4, 4, 8, 8, 9, 12, 12, 14,
16, 23, 85.

Алгоритм и особенности решения задачи.

- Для начала надо определиться, как у нас будут формироваться первые два массива. Использование **random** нам не подходит, т.к. массивы должны быть упорядоченными. Будем вводить данные с клавиатуры.
- Длина третьего массива равна сумме двух первых.
- Элементы под текущими для каждого массива индексами сравниваются, и меньший по значению записывается в третий массив. При этом индекс массива, из которого элемент был скопирован в третий массив, должен увеличиться.
- Может возникнуть ситуация, когда один массив закончился, а второй еще нет. Поэтому в программе должен быть организован цикл записи возможного окончания как первого так и второго массива.


```
const N = 7;  
var arr1, arr2: array[1..N] of integer;  
arr3: array[1..N+N] of integer;  
i, j, k, m: byte;  
begin  
  writeln('Заполните первый массив:');  
  for i := 1 to N do read(arr1[i]);  
  writeln('Заполните второй массив:');  
  for i := 1 to N do read(arr2[i]);
```

```
i := 1; j := 1; k := 1;
  while (i <= N) and (j <= N) do
begin
  if arr1[i] < arr2[j] then
    begin arr3[k] := arr1[i]; i := i + 1 end
    else begin arr3[k] := arr2[j]; j := j + 1 end;
  k := k + 1;
end;
```

while $i \leq N$ **do**

begin

$arr3[k] := arr1[i];$

$i := i + 1;$

$k := k + 1$

end;

while $j \leq N$ **do begin**

$arr3[k] := arr1[j]; j := j + 1; k := k + 1$

end;

writeln;

for $i := 1$ **to** $N+N$ **do** write(arr3[i]:4);

end.