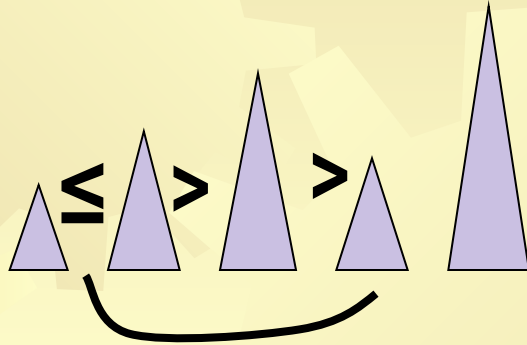
The background features a light yellow-to-gold gradient with several semi-transparent gear shapes scattered across it. On the far left, there is a vertical strip with a colorful, abstract, and somewhat pixelated pattern in shades of red, orange, purple, and blue.

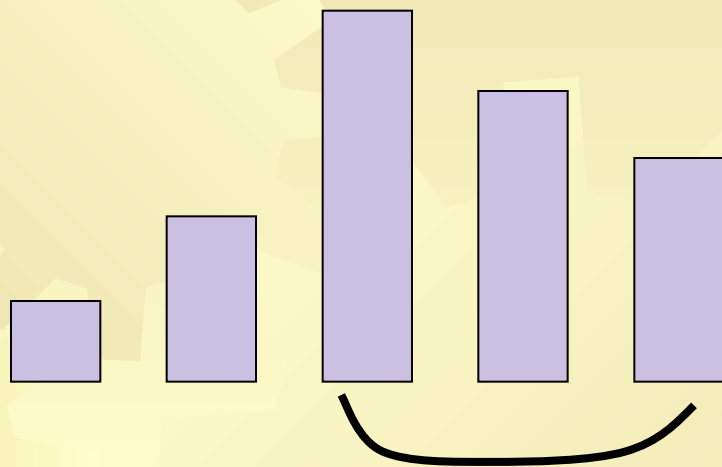
# **Сортировка данных в массиве**

# Метод вставок



- ☀ Место любого элемента последовательности выбирается так, чтобы он стоял после элемента, который меньше или равен ему.
- ☀ Часть последовательности должна сдвинуться на 1 позицию вправо.

# Метод выбора



- ✦ В неупорядоченной последовательности отыскивается минимальный элемент, и он меняется местами с 1 элементом неупорядоченной части.

```
PROGRAM VYBOR;  
VAR I,J,N,M,MIN,MIT: INTEGER;  
    A:ARRAY[1..100] OF INTEGER;  
BEGIN  
WRITE('N');  
READLN (N);  
{Заполнение и распечатка массива}  
FOR I:=1 TO N DO A[I]:=RANDOM(99)+1;  
FOR I:=1 TO N DO WRITE (A[I]:3);  
WRITELN;
```

{Поиск минимального элемента массива}

FOR J:=1 TO N-1 DO

BEGIN

MIN:=A[J];

M:=J;

FOR I:=J+1 TO N DO

IF A[I]<MIN THEN

BEGIN

MIN:=A[I];

M:=I;

END;

{Обмен элементов местами}

MIT:=A[J];

A[J]:=MIN;

A[M]:=MIT;

END;

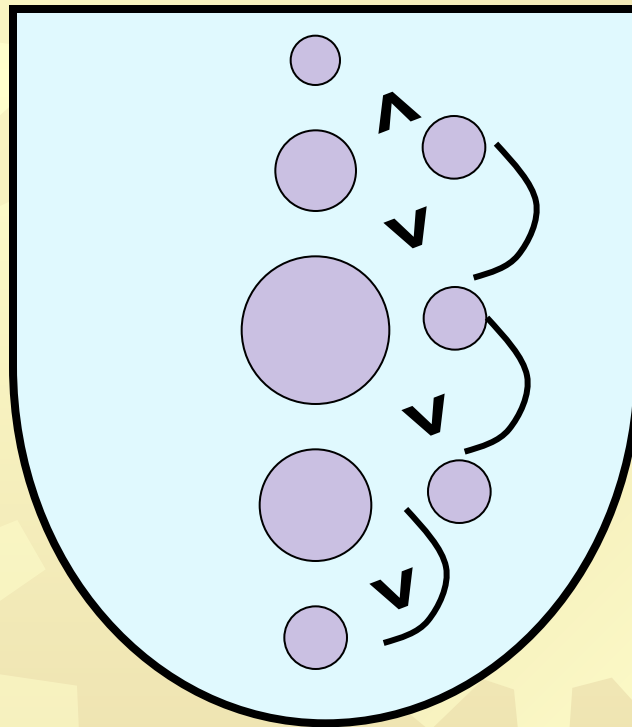
{Распечатка упорядоченного массива}

FOR I:=1 TO N DO WRITE (A[I]:3);

END.

# Метод обмена (пузырька)

- ✦ Состоит в последовательных просмотрах массива от конца к началу, и обмене местами соседних элементов.



```
PROGRAM BUBLE UP;  
VAR I,J,N,MIT: INTEGER;  
    A:ARRAY[1..100] OF INTEGER;  
BEGIN  
WRITE('N');  
READLN (N);  
{Заполнение и распечатка массива}  
FOR I:=1 TO N DO A[I]:=RANDOM(99)+1;  
FOR I:=1 TO N DO WRITE (A[I]:3);  
WRITELN;
```





{Сортировка}

FOR I:=N DOWNTO 2 DO

  BEGIN

    J:=I;

    WHILE (J>=2) DO

      BEGIN

        IF A[J-1]>A[J] THEN

          BEGIN

            MIT:=A[J-1];

            A[J-1]:=A[J];

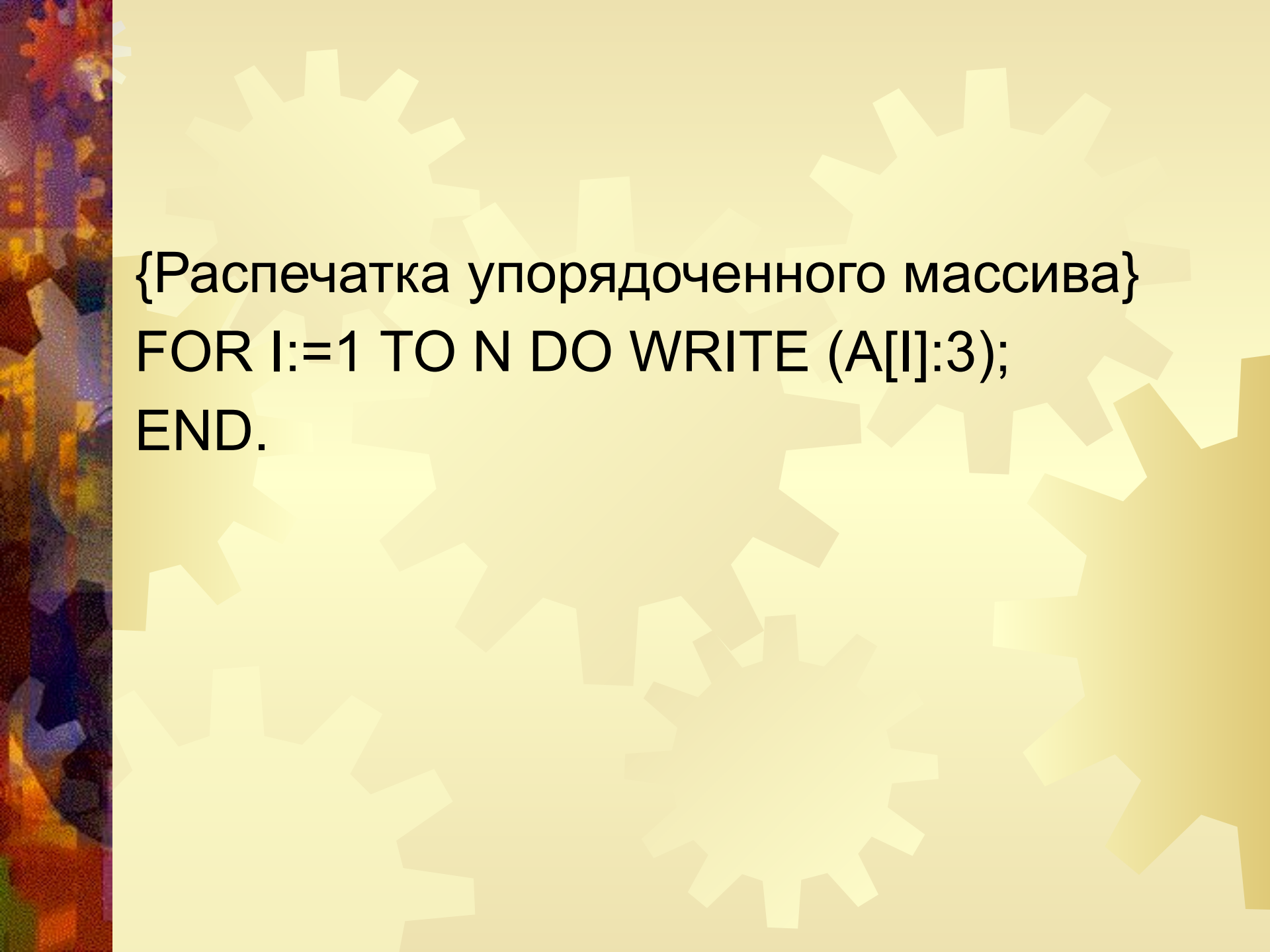
            A[J]:=MIT;

          END;

        J:=J-1;

      END;

  END;



{Распечатка упорядоченного массива}  
FOR I:=1 TO N DO WRITE (A[I]:3);  
END.