

МАССИВЫ

(примеры задач)

Сортировка пузырьком

```
program bubble_sort;
const n=6;
var a: array[1..n] of real; x: real;
    i,j: integer;
begin
    writeln('input array');
    for i:=1 to n do read (a[i]);
    readln;
    for i:= 1 to n-1 do
        for j:=1 to n-i do
            if a[j]>a[j+1]
            then
                begin
                    x:=a[j]; a[j]:= a[j+1]; a[j+1]:=x;
                end;
        writeln('result:');
        for i:=1 to n do write (a[i]:6:2);
        readln;
    end.
```

Сортировка выбором

```
program StraightSelection;
var i,j,k: index; x: item;
begin
  for i:=1 to n-1 do
    begin
      k:=i; x:=a[i];
      for j:=i+1 to n do
        if a[j] < x
        then
          begin
            k:=j; x:=a[j]
          end;
      a[k]:=a[i]; a[i]:=x;
    end;
end; {StraightSelection}
```

Очень важный пример!

Текст не более одной строки.
Подсчитать количество вхождений
малых латинских букв. Результат
распечатать по алфавиту

Все дело в структуре данных!

Она решает все!

```
const
    max_length=80;
type
    Tletter = 'a'..'z'; //вариант задачи: #0..#255
var
    txt: array[1..max_length] of char;
    counters: array[Tletter] of integer;
    i,n: integer;
    c: char;
```

```

program letters;
const
    max_length=80;
type
    Tletter = 'a'..'z';
var
    txt: array[1..max_length] of char;
    counters: array[Tletter] of integer;
    i,n: integer;
    c: char;
begin
    i:=0;
    writeln('Введите текст:');
    repeat
        i:=i+1;
        read(txt[i]);
    until eoln or (i=max_length);
    readln;
    n:=i;
    for c:='a' to 'z' do //инициализация счетчиков
        counters[c]:=0;
    for i:=1 to n do
        begin
            c:=txt[i];
            if (c>='a') and (c<='z')
            then
                counters[c]:= counters[c]+1; //counters[txt[i]]
            end;
        writeln('частота встречаемости:');
        for c:='a' to 'z' do
            write(c, '-', counters[c], ' ');
        readln;
    end.

```

Решето Эратосфена

Решето Эратосфена - классический алгоритм нахождения простых чисел (пусть надо найти меньше 10 чисел).

2 3 4 5 6 7 8 9 10

Уберем самое маленькое число, решив, что оно простое, и все числа, делящиеся на него.

После первого шага будем иметь простое число 2 и решето, содержащее только нечетные числа 3 5 7 9.

Аналогично поступаем с 3, в решете останутся 5 и 7.

Процесс закончится при опустошении решета.

```

program prime_numbers;
const
  first_prime_number=2;
  max=10000;
var
  {решето Эратосфена: sieve [siv] of Eratosthenes}
  sieve: array[first_prime_number.. max] of boolean;
  range {диапазон},
  i,j: integer;

begin
  write('Диапазон?'); readln(range);
  for i:=first_prime_number to range do
    sieve[i]:=true;
  for i:=first_prime_number to range-1 do
    if sieve[i] then
      for j:=i+1 to range do
        if j mod i=0
          then sieve[j]:=false;
  writeln('Простые числа');
  for i:=first_prime_number to range do
    if sieve[i] then write (i:6);
  readln;
end.

```