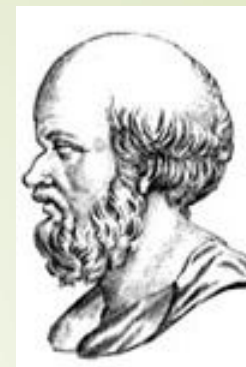


# Алгоритмизация и программирование

1

§ 38. Целочисленные алгоритмы

# Решето Эратосфена



Эратосфен Киренский  
(Eratosthenes, Ερατοσθένης)  
(ок. 275-194 до н.э.)

## Алгоритм:

- 1) начать с  $k = 2$
- 2) «выколоть» все числа через  $k$ , начиная с  $k \cdot k$
- 3) перейти к следующему «невыколотому»  $k$
- 4) если  $k \cdot k \leq N$ , то перейти к шагу 2
- 5) напечатать все числа, оставшиеся «невыколотыми»

Новая версия – решето Аткина.

**?** Как улучшить?

**+** высокая скорость, количество операций

$$O((N \cdot \log N) \cdot \log \log N)$$

**-** нужно хранить в памяти все числа от 1 до  $N$

# Решето Эратосфена

Задача. Вывести все простые числа от 2 до  $N$ .

Объявление переменных:

```
цел i, k, N = 100  
логтаб A[2:N]
```

```
const N = 100;  
var i, k: integer;  
    A: array[2..N]  
      of boolean;
```

Сначала все невычеркнуты:

```
нц для i от 2 до N  
    A[i] := да  
кц
```

```
for i := 2 to N do  
    A[i] := True;
```

# Решето Эратосфена

## Вычёркивание непростых:

```
к := 2
нц пока  $k*k \leq N$ 
  если  $A[k]$  то
     $i := k*k$ 
    нц пока  $i \leq N$ 
       $A[i] := \text{нет}$ 
       $i := i + k$ 
    кц
  все
   $k := k + 1$ 
кц
```

```
к := 2;
while  $k*k \leq N$  do begin
  if  $A[k]$  then begin
     $i := k*k$ ;
    while  $i \leq N$  do begin
       $A[i] := \text{False}$ ;
       $i := i + k$ 
    end
  end;
   $k := k + 1$ 
end;
```

# Решето Эратосфена

---

## Вывод результата:

```
нц для i от 2 до N
  если A[i] то
    вывод i, нс
  все
кц
```

```
for i:=2 to N do
  if A[i] then
    writeln ( i );
```