

# Разбор задач муниципального этапа олимпиады по информатике

## 1. Рассеянный математик (все классы)

Тема: разбор случаев

Сложность: простая

1-е число является одним из размеров какой-то комнаты, другой размер этой комнаты может быть 2-м, 3-м или 4-м числом. Два других числа являются размерами второй комнаты. Таким образом нужно рассмотреть три случая и выбрать максимальное значение из следующих вариантов для суммарной площади комнат:  $a*b+c*d$ ,  $a*c+b*d$  и  $a*d+b*c$  (буквами  $a,b,c,d$  обозначены четыре числа в порядке ввода).

Пример реализации:

```
var s,ms,a,b,c,d:integer;
begin
  read(a,b,c,d);
  ms:=a*b+c*d;
  s:=a*c+b*d;
  if s>ms then
    ms:=s;
  s:=a*d+b*c;
  if s>ms then
    ms:=s;
  writeln(ms);
end.
```

## 2. Блок-схема (8 класс)

Тема: реализация программы по блок-схеме

Сложность: простая

Пример реализации:

```
var n,k:integer;
begin
  read(n);
  k:=2;
  while k*k<=n do
  begin
    while n mod k=0 do
    begin
      writeln(k);
      n:=n div k;
    end;
    k:=k+1;
  end;
  if n>1 then
    writeln(n);
end.
```

## 2. Муравей (9-11 класс)

Тема: поиск максимума

Сложность: простая

Сначала нужно найти номер  $m$  первого максимального элемента в массиве высот. В массив высот  $h_i$  можно добавить вспомогательный элемент  $h_0=0$ . Длина пути муравья будет равна сумме  $|h_i - h_{i-1}|$  для  $i$  от 1 до  $m$  (по вертикали) плюс  $m - 1$  (по горизонтали).

### 3. Изменения температуры (8 класс)

Тема: поиск максимума

Сложность: простая

Требуется найти номер  $i$  в диапазоне от 2 до  $N$  при котором значение  $|a_i - a_{i-1}|$  максимально (здесь  $a_i$  – показания термометра). Скобками  $||$  обозначается модуль (абсолютное значение) числа.

При реализации программы можно обойтись без массива, сохраняя предыдущее введенное значение в переменной  $t1$ :

```
var mdt, dt, mi, i, t1, t2, n: integer;
begin
  read(n);
  read(t1);
  mdt:=0;
  mi:=2;
  for i:=2 to n do
  begin
    read(t2);
    dt:=abs(t2-t1);
    if dt>mdt then
    begin
      mdt:=dt;
      mi:=i;
    end;
    t1:=t2;
  end;
  writeln(mi);
end.
```

### 3. Теорема Васи (9-11 класс)

Тема: модулярная арифметика, быстрое возведение в степень

Сложность: средняя

При решении задачи необходимо воспользоваться алгоритмом быстрого возведения в степень, который подробно разобран в разборе задачи 1500. «Вундеркинд», предлагавшейся на отборочных командных соревнованиях в 2010 году.

### 4. Сумма (8 класс)

Тема: математика

Сложность: средняя

Реализация алгоритма в лоб дает только 50 баллов, так как не проходит ограничения по времени при больших  $N$ . Можно заметить, что в некотором диапазоне частное будет одинаковым и равно  $k$ .

Максимальным значением  $j$ , при котором частное будет равно  $k$ , является  $\lfloor \frac{N}{k} \rfloor$ . Таким образом

суммирование нужно производить диапазонами от  $i$  до  $j$ , в котором частное совпадает. Кроме того, для суммирования нужно использовать переменную вещественного типа или `int64`, чтобы не происходило переполнения. Пример реализации:

```
var i, j, n: longint;
    s, k: int64;
begin
  read(n);
  i:=1;
  s:=0;
  while i<=n do
  begin
    k:=n div i;
    j:=n div k;
    s:=s+k*(j-i+1);
    i:=j+1;
  end;
  writeln(s);
end.
```

## 4. Двоичные числа (9-11 класс)

Тема: алгоритм слияния возрастающих последовательностей

Сложность: выше среднего

Последовательность чисел можно задать рекуррентно через слияние последовательностей, получаемых при умножении этой же последовательности на двоичные числа 2, 22, 222, ..., 22...22 (18 двоек).

Первый элемент последовательности равен 1. Каждая вспомогательная сливаемая последовательность определяется номером элемента основной последовательности, использованным для получения очередного элемента вспомогательной последовательности (первоначально все индексы установлены на 1) и значением текущего элемента. При слиянии последовательностей нужно найти минимальное значение из текущих элементов вспомогательных последовательностей — это и станет очередным элементом основной последовательности. Затем нужно сместить индексы и пересчитать текущие элементы в тех вспомогательных последовательностях, текущие элементы которых совпадают с использованным минимальным значением (таких последовательностей может быть несколько).

Для умножения нужно определить функцию, которая будет возвращать очень большое число (например, 22...22 (18 двоек)) в случае переполнения. Также для вычислений нужно использовать тип int64. Пример реализации:

```
var x,y:int64;
    d:array[1..18]of int64; { двоичные числа }
    e:array[1..18]of int64; { текущие элементы }
    si:array[1..18]of int64; { индексы }
    s:array[1..10000]of int64; { результирующая последовательность }
    i,k,m:integer;
function mul(a,b:int64):int64; { умножение с проверкой на переполнение }
var r:int64;
begin
    r:=a*b;
    if r div b<>a then r:=d[18];
    mul:=r;
end;
begin
    read(x);
    { инициализация }
    d[1]:=2;
    for i:=2 to 18 do
        d[i]:=d[i-1]*10+2;
    s[1]:=1;
    for i:=1 to 18 do
    begin
        si[i]:=1;
        e[i]:=d[i];
    end;
    k:=1;
    { слияние }
    while s[k]<x do
    begin
        m:=1;
        for i:=2 to 18 do { поиск минимума }
            if e[m]>e[i] then
                m:=i;
        y:=e[m];
        inc(k);
        s[k]:=y;
        for i:=1 to 18 do
            if e[i]=y then { смещение индексов и пересчет текущих элементов }
            begin
                inc(si[i]);
                e[i]:=mul(s[si[i]],d[i]);
            end;
        end;
        writeln(s[k]);
    end.
```