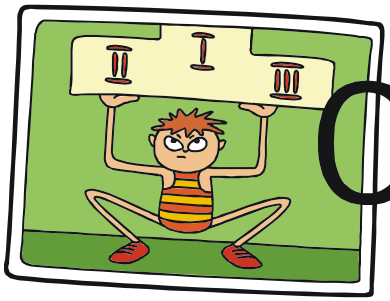


# Олимпиады (без допинга)

***Смыкалов Владимир Павлович,***
***Замятин Евгений Игоревич,***
***Ульянцев Владимир Игоревич***

## ЗАДАЧА «ПРОГУЛКА ПО ПАРКУ»

Этой статьей мы продолжаем цикл публикаций олимпиадных задач по информатике для школьников. Решение таких задач поможет вам повысить уровень практических навыков программирования и подготовиться к олимпиадам по информатике.

В данной статье рассматривается задача «Прогулка по парку», которая предлагалась на Третьей личной интернет-олимпиаде по программированию в 2014/15 учебном году. Материалы олимпиады можно найти на сайте http://neerc.ifmo.ru/school/io/.

### УСЛОВИЕ ЗАДАЧИ

Последняя неделя выдалась дождливой. Мистер Бин вдоволь насиделся дома и в этот солнечный день решил пойти погулять в центральный парк. Идя по аллее, мистер Бин увидел красочную вывеску, которая гласила: «2015 год – год света!». Эта надпись подняла настроение нашему герою и он, полный вдохновения, направился к месту назначения творить добрые дела!

Придя в парк, он увидел скульптуру, представляющую из себя совокупность ваз разных размеров. Ваза с номером $i$ имела размер $a_i$, и за неделю в ней накопилось $a_i$ единиц воды. У Мистера Бина было озорное настроение, поэтому в каждой вазе он проделал отверстие. Отверстие в каждой вазе имеет свой размер: из $i$-ой вазы в секунду вытекает $b_i$ единиц воды. Сделав доброе дело, мистер Бин заинтересовался, сколько суммарно воды во всех вазах находится в каждый момент времени по прошествии от 0 до $t$ секунд после момента появления отверстий. Считается, что отверстия появились во всех вазах одновременно.

Формат входного файла

В первой строке входного файла даны два числа $n$, $t$ ($1 \le n \le 10^5$, $1 \le t \le 10^6$) – число ваз и время в секундах, в течение которого мистера Бина интересовала вода в вазах. В каждой следующей строке даны два числа $a_i$, $b_i$ ($0 \le a_i \le 10^9$, $0 \le b_i \le 10^9$) – описание $i$-й вазы.

Формат выходного файла

В качестве ответа выведите $t + 1$ целое число. Первая строка должна содержать суммарное количество воды в начальный мо-

мент. В каждой следующей строке выведите суммарное количество воды в очередной момент.

Примеры входных и выходных данных

| vase.in | vase.out |
|---|---|
| 6 6<br>12 2<br>10 3<br>3 1<br>5 4<br>7 2<br>9 1 | 46<br>33<br>23<br>14<br>9<br>6<br>3 |

## РАЗБОР ЗАДАЧИ

Будем решать задачу поэтапно: сперва придумаем простое решение, а затем будем его улучшать. Рассмотрим случай с одной вазой и посмотрим, как меняется количество воды в ней. Предположим, что изначально в ней $a$ единиц воды, а сквозь дырочку за секунду вытекает $b$ единиц воды. Тогда выполняется следующая зависимость (см. табл. 1).

Количество воды в вазе будет уменьшаться по $b$ до определенного момента, а в последующий момент воды в вазе не останется. Нетрудно заметить, что $k$ – это частное от деления $a$ на $b$ ($k = a$ **div** $b$). На последнем шаге, после $k$-го момента времени, количество воды в вазе уменьшится на $a - kb = a$ **mod** $b$.

Рассмотрим такое решение для случая с $n$ вазами: будем в каждый момент времени хранить суммарное количество воды в вазах и суммарную скорость, с которой вода вытекает из ваз. Сначала подсчитаем изначальное количество воды во всех вазах и изначальную скорость вытекания.

Программная реализация на языке *Pascal* приведена в листинге 1.

Теперь пройдемся циклом для каждого момента времени от 0 до $t$, подсчитаем количество воды в этот момент времени и не забудем его вывести (листинг 2).

**Табл. 1**

| Момент времени | 0 | 1 | 2 | ... | $k$ | $k+1$ | ... |
|---|---|---|---|---|---|---|---|
| Количество воды | $a$ | $a-b$ | $a-2b$ | ... | $a-kb$ | 0 | ... |

**Листинг 1.** Подсчет sumA и sumB

```
sumA := 0;
sumB := 0;
for i := 1 to n do begin
  sumA := sumA + a[i];
  sumB :=
```

**Листинг 2.** Основной цикл программы

```
writeln(sumA);
for i := 1 to t do begin
  for j := 1 to n do begin
    if (a[j] div b[j] = i - 1) then begin
      sumA := sumA - a[j] mod b[j];
      sumB := sumB - b[j];
    end;
  end;
  sumA := sumA - sumB;
  writeln(sumA);
end;
```

Приведенное решение уже работает на тесте из условия! Теперь исправим недочеты и ускорим решение. Сначала заметим, что `b[j]` может равняться нулю, а делить на 0 нельзя. Можно считать, что в этом случае скорость вытекания воды из вазы никогда не меняется. Исправим эту строчку, добавив условие `b[j] > 0` в соответствующий условный оператор **`if`**.

Теперь решение работает на всех тестах с $n \le 100$, что уже дало бы 40 баллов из 100 на олимпиаде. Оценим время работы данного алгоритма. Нетрудно заметить, что время его работы составляет $O(n^2)$. С таким временем работы алгоритм не уложится по времени, например, при $n = 100000$.

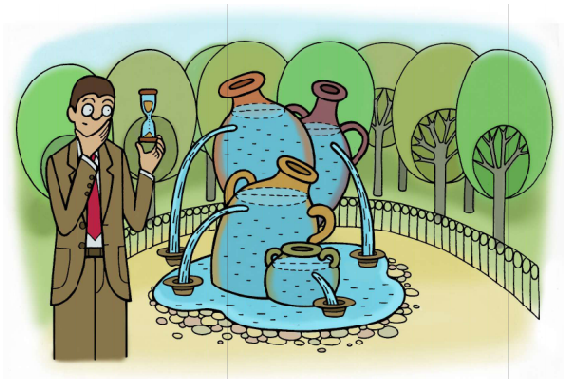
Заметим, что строки внутри **`if`** выполняются для каждого $j$ только один раз, а значит, есть возможность значительно сократить время работы программы. Сначала для удобства создадим структуру данных типа **`record`** для вазы (листинг 3).

В данном листинге `a` и `b` – первоначальное количество воды в вазе и скорость вытекания, так же как и в условии. Добавим еще параметр `t = a div b`, равный номеру момента времени, после которого ничего в вазе не останется. В случае `b = 0` будем считать, что `t = 1e9`.

Представим, что мы отсортировали все вазы по возрастанию параметра `t`. Тогда нам нужно обрабатывать вазы согласно их порядку. А значит, алгоритм можно будет переписать: новая версия приведена в листинге 4.

Теперь указанная часть алгоритма работает за время $O(n)$. Остался последний шаг: быстро отсортировать все вазы по возрастанию параметра `t`. Для этого воспользуемся алгоритмом быстрой сортировки, который работает за $O(n\ log(n))$. Подробнее про этот алгоритм можно прочесть в [1] или [2], мы лишь приведем пример его несложной реализации (листинг 5).

Наконец, нужно не забыть поставить тип **`int64`** у переменных `sumA` и `sumB` (ведь значения в них могут достигать $10^{15}$), и решение пройдет все тесты!



**Листинг 3.** Структура данных для вазы

```
type
  Vase = record
    a, b, t: longint;
  end;
```

**Листинг 4.** Исправленный основной цикл

```
writeLn(sumA);
pos := 1;
for i := 1 to t do begin
    while (pos <= n) and (vases[pos].t = i - 1) do begin
        sumA := sumA - vases[pos].a mod vases[pos].b;
        sumB := sumB - vases[pos].b;
        inc(pos);
    end;
    sumA := sumA - sumB;
    writeLn(sumA);
end;
```

**Листинг 5.** Быстрая сортировка

```
procedure quickSort(left, right: longint);
var
  i, j: longint;
  tmp, mid: Vase;
begin
  i := left;
  j := right;
  mid := vases[(left + right) shr 1];
  repeat
    while mid.t > vases[i].t do
      Inc(i);
    while mid.t < vases[j].t do
      Dec(j);
    if i <= j then begin
      tmp := vases[i];
      vases[i] := vases[j];
      vases[j] := tmp;
      Inc(i);
      Dec(j);
    end;
  until i > j;
  if left < j then
    quickSort(left, j);
  if i < right then
    quickSort(i, right);
end;
```

**Источники:**

1. *Скиена С.* Алгоритмы. Руководство по разработке. СПб: БХВ-Петербург, 2011.
2. *Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К.* Алгоритмы. Построение и анализ. М.: Вильямс, 2013.

***Смыкалов Владимир Павлович,***
***студент пятого курса кафедры «Компьютерные технологии» Университета ИТМО,***

***Замятин Евгений Игоревич,***
***студент четвертого курса кафедры «Компьютерные технологии» Университета ИТМО,***

***Ульянцев Владимир Игоревич,***
***кандидат технических наук, доцент кафедры «Компьютерные технологии» Университета ИТМО, член жюри Интернет-олимпиад по информатике.***