

Турнир Архимеда 2013



Командная олимпиада по программированию

Задача А. Аудиозаписи

- Рассмотрим число, у которого все цифры различные и идут в возрастающем порядке:

$$A = a_1 a_2 a_3 \dots a_n, \quad \underline{a_1} < \underline{a_2} < \dots < \underline{a_n}$$

- $9 * A = 10 * A - A = a_1 a_2 a_3 \dots a_n 0 - a_1 a_2 a_3 \dots a_n$

Задача А. Аудиозаписи

- Теперь рассмотрим подробнее на примере пятизначного числа

a_1	a_2	a_3	a_4	a_5	0
	a_1	a_2	a_3	a_4	a_5
a_1	$a_2 - a_1$	$a_3 - a_2$	$a_4 - a_3$	$a_5 - 1 - a_4$	$10 - a_5$

- Поскольку цифры шли в возрастающем порядке, все полученные разности будут неотрицательными.
- Это и будут цифры числа $9 \cdot A$

Задача А. Аудиозаписи

Найдем сумму цифр полученного числа:

$$a_1 + a_2 - a_1 + a_3 - a_2 + a_4 - a_3 + a_5 - 1 - a_4 + 10 - a_5 =$$

$$\cancel{a_1} + \cancel{a_2} - \cancel{a_1} + \cancel{a_3} - \cancel{a_2} + \cancel{a_4} - \cancel{a_3} + \cancel{a_5} - 1 - \cancel{a_4} + 10 - \cancel{a_5} =$$

$$10 - 1 = 9$$

Таким образом, независимо от входных данных, ответ в задаче - 9.

Задача В. Пробка

- Самый первый автомобиль движется со своей максимальной скоростью.
- Для каждого следующего автомобиля его скорость равна минимуму из его максимальной скорости и скорости автомобиля, едущего перед ним.
- Циклом выводим все скорости.

Задача В. Пробка

Пример кода:

```
var
  n, i, prev, cur: integer;
begin
  readln(n);
  prev := 300; //поскольку все скорости не
                //больше 300
  for i := 1 to n do begin
    read(cur);
    prev := min(prev, cur);
    write(prev, ' ');
  end;
end.
```

Задача С. Красивые номера

Для начала научимся
по букве понимать её
цифру

1	2 abc	3 def
4 ghi	5 jkl	6 mno
7 pqrs	8 tuv	9 wxyz

Задача С. Красивые номера

```
function letter_digit(letter : char) : integer
begin
    if letter < 'd' then
        letter_digit := 2
    else if letter < 'g' then
        letter_digit := 3
    else if letter < 'j' then
        letter_digit := 4
    else if letter < 'm' then
        letter_digit := 5
    else if letter < 'p' then
        letter_digit := 6
    else if letter < 't' then
        letter_digit := 7
    else if letter < 'w' then
        letter_digit := 8
    else
        letter_digit := 9;
end;
```


Задача С. Красивые номера

- Теперь по названию компании будем определять её номер:

```
function phone_num(var company : string) :  
integer;  
var  
    num, i : integer;  
begin  
    num := 0;  
    for i := 1 to 7 do  
        num := (num * 10) + letter_digit(company  
[i]);  
    phone_num := num;  
end;
```

Задача С. Красивые номера

- Изначально будем предполагать, что все фирмы могут получить красивые номера.
- `ans = N;`
- Как только будем находить пару фирм, у которых должен быть одинаковый номер, будем уменьшать ответ.
- Для того, чтобы знать, какие номера ещё свободны,ведем массив
- `free : array[2222222..9999999] of boolean;`
- Изначально заполняем его `true`

Задача С. Красивые номера

- По названию каждой компании узнаем её красивый номер. Если этот номер уже занят, то уменьшаем ответ. Если номер свободен, то "занимаем" его.
- ```
for i := 1 to n do
begin
 readln(companyName);
 currNum := phone_num(companyName);

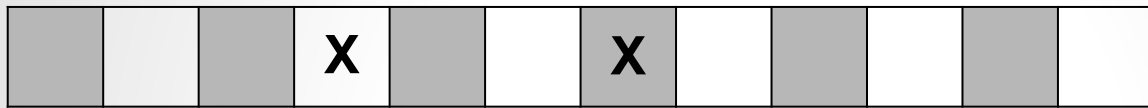
 if (Empty[currNum]) then
 Empty[currNum] := false
 else
 dec(ans);
end;
```
- Выводим ответ.

# Задача D. Доминошахматы

- Для начала проверяем, что доска имеет четную площадь.
- Для того, чтобы доску можно было замостить, необходимо, чтобы вырезанные клетки были разных цветов.
  - Доминошка покрывает 2 соседние клетки, т.е. всегда клетки разных цветов.
  - В исходной доске четной площади черных и белых клеток одинаковое количество.
  - Если вырезали две клетки одного цвета, то количество черных не равно количеству белых. Замостить нельзя.
- Проверка на то, что цвета разные:
  - суммы координат каждой из клеток дают разные остатки при делении на 2
  - $(x_1 + y_1) \bmod 2 \neq (x_2 + y_2) \bmod 2$

# Задача D. Dominoшахматы

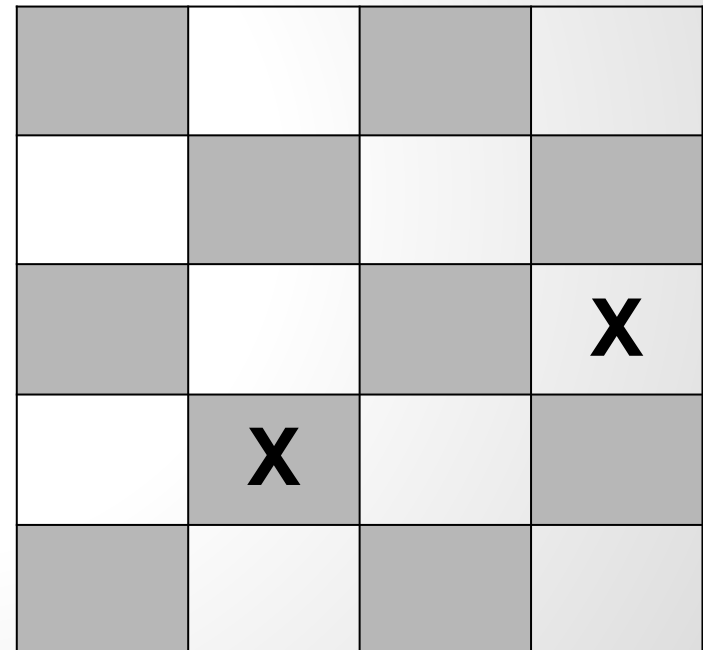
- Того, что вырезанные клетки разного цвета, недостаточно.
- Случай "полоска  $1 \times N$ ",  $N$  - четное



- В приведенном примере нельзя замостить, поскольку кусочки слева и справа от вырезанных клеток нечетной длины.
- Нужно проверить, что все три кусочка, на которые две вырезанные клетки разбивают полоску, имеют четную длину(если вырезанные клетки идут подряд, то считаем, что между ними кусок длины 0)

# Задача D. Dominoшахматы

- Пусть
  - доска четной площади (минимум одна сторона четная),
  - вырезанные клетки разного цвета
  - не полоска.
- Построим замощение



# Задача D. Доминошахматы

- Будем идти змейкой, начиная вдоль четной стороны (для удобства пусть это ширина таблицы)
- Идем из левого нижнего угла направо -- вверх -- налево -- вверх и т.д.
- Будем стараться ставить "горизонтальные" доминошки, если не получается, то "вертикальную".

|   |          |   |          |
|---|----------|---|----------|
| 8 | 8        | 9 | 9        |
| 7 | 7        | 6 | 6        |
| 4 | 5        | 5 | <b>X</b> |
| 4 | <b>X</b> | 3 | 3        |
| 1 | 1        | 2 | 2        |

# Задача D. Доминошахматы

- На нашем пути между вырезанными клетками всегда четное число клеток.

## Доказательство.

Вырезанные клетки разного цвета.

При переходе из клетки в любую соседнюю мы меняем цвет.

Всего мы сменим цвет нечетное число раз. Значит, перейдем нечетное число границ клеток, а самих клеток посетим четное

|   |   |   |   |
|---|---|---|---|
| 8 | 8 | 9 | 9 |
| 7 | 7 | 6 | 6 |
| 4 | 5 | 5 | X |
| 4 | X | 3 | 3 |
| 1 | 1 | 2 | 2 |



# Задача D. Доминошахматы

- Теперь рассмотрим кусочки до первой вырезанной и после второй.
- В нашем случае они оба четной длины, поэтому построить замощение просто.
- Пусть это не так, т.е. оба кусочка нечетной длины.

|   |   |   |   |
|---|---|---|---|
| 8 | 8 | 9 | 9 |
| 7 | 7 | 6 | 6 |
| 4 | 5 | 5 | X |
| 4 | X | 3 | 3 |
| 1 | 1 | 2 | 2 |

# Задача D. Доминошахматы

- Если заполнять так, как мы делали раньше, то не получается.
- Рассмотрим ряд таблицы содержащий вырезанную клетку. Поскольку длина этого ряда четная, то с одной из сторон от вырезанной клетки будет четное число клеток, а с другой нечетное.
- Сделаем заполнение змейкой таким, чтобы приходить к вырезанной клетке со стороны с четным количеством.
- Для этого изменим направление заполнения

|   |          |          |     |
|---|----------|----------|-----|
| 7 | 8        | 8        | ??? |
| 7 | <b>X</b> | 6        | 6   |
| 4 | 4        | 5        | 5   |
| 3 | 3        | <b>X</b> | ??? |
| 1 | 1        | 2        | 2   |

# Задача D. Доминошахматы

Новый вариант

|   |   |   |   |
|---|---|---|---|
|   |   |   |   |
|   | X |   |   |
|   |   |   |   |
| 3 | 3 | X |   |
| 2 | 2 | 1 | 1 |



Старый вариант

|   |   |   |     |
|---|---|---|-----|
| 7 | 8 | 8 | ??? |
| 7 | X | 6 | 6   |
| 4 | 4 | 5 | 5   |
| 3 | 3 | X | ??? |
| 1 | 1 | 2 | 2   |



# Задача D. Доминошахматы

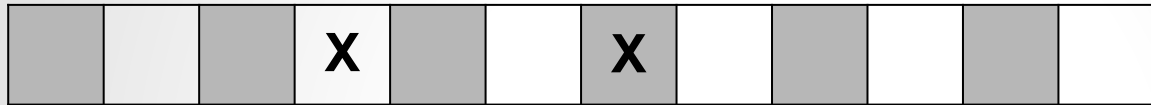
- Поскольку "до" первой вырезанной клетки кусок четной длины и между вырезанными клетками тоже кусок четной длины, оставшихся клеток тоже будет четное число.
- Значит, их можно замостить

|   |   |   |   |
|---|---|---|---|
| 9 | 9 | 8 | 8 |
| 6 | X | 7 | 7 |
| 6 | 5 | 5 | 4 |
| 3 | 3 | X | 4 |
| 2 | 2 | 1 | 1 |

# Задача D. Доминошахматы

## *Второй способ.*

- Того, что вырезанные клетки разного цвета, недостаточно.
- Случай "полоска  $1 \times N$ ",  $N$  - четное



- В приведенном примере нельзя замостить, поскольку кусочки слева и справа от вырезанных клеток нечетной длины.
- Нужно проверить, что все три кусочка, на которые две вырезанные клетки разбивают полоску, имеют четную длину (если вырезанные клетки идут подряд, то считаем, что между ними кусок длины 0)
- Заметим, что если две клетки разного цвета, то центральный кусок четный
- Если первый кусок четный, то тогда и последний кусок четный.
- Таким образом нам надо проверить, что только первый кусок четный

# Задача D. Доминошахматы

- Вопрос: что делать с большой доской!
- Ответ: свести к полоске!
- Будем обходить доску змейкой
- Заметим, что если номера двух клеток отличаются на единицу, то мы можем покрыть их одной доминошкой.

|    |    |    |    |
|----|----|----|----|
| 17 | 18 | 19 | 20 |
| 16 | 15 | 14 | 13 |
| 9  | 10 | 11 | 12 |
| 8  | 7  | 6  | 5  |
| 1  | 2  | 3  | 4  |

# Задача D. Доминошахматы

- После такой нумерации если мы замостим полосу  $1 \times NM$ , то мы замостим и поле  $N \times M$
- Если в замощении полосы доминошка покрывает клетки  $i$  и  $i+1$ , то она накроет их же в замощении поля  $N \times M$

|    |    |    |    |
|----|----|----|----|
| 17 | 18 | 19 | 20 |
| 16 | 15 | 14 | 13 |
| 9  | 10 | 11 | 12 |
| 8  | 7  | 6  | 5  |
| 1  | 2  | 3  | 4  |

|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |
|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|

# Задача D. Доминошахматы

- Для того, чтобы замостить полосу, надо чтобы первый кусок был четным. Это значит, что номер первой вырезанной клетки -- нечетный.
- Но мы можем запустить змейку в первом ряду справа налево, а можем слева направо.
- Если мы запустим змейку вдоль четной стороны, то в этих двух способах номера первой вырезанной клетки будут разной четности. То есть в одном из случаев он будет нечетным.
- Победа!

|    |    |    |    |
|----|----|----|----|
| 17 | 18 | 19 | 20 |
| 16 | 15 | 14 | 13 |
| 9  | 10 | 11 | 12 |
| 8  | 7  | 6  | 5  |
| 1  | 2  | 3  | 4  |

|    |    |    |    |
|----|----|----|----|
| 20 | 19 | 18 | 17 |
| 13 | 14 | 15 | 16 |
| 12 | 11 | 10 | 9  |
| 5  | 6  | 7  | 8  |
| 4  | 3  | 2  | 1  |



# Задача Е. Слишком много кофеина

- Заметим, что буква "Р" есть только в числах 3 и 4 и их производных: 30, 40, 300, 400 и т. д.
- Значит, в том числе, которое надо найти, не должно быть цифр 3 и 4.
- Будем увеличивать число, сказанное Ниной, на 1, пока не получим "хорошее". Оно и будет нашим ответом.
- ```
begin
    read(n);
    inc(n);
    while not correct(n) do inc(n);
    writeln(n);
end.
```

Задача Е. Слишком много кофеина

- Проверка числа на корректность:

```
function correct(n : integer) : boolean;  
var  
    t : string;  
    i : integer;  
begin  
    t := inttostr(n);  
    correct := false;  
    for i := 1 to length(t) do  
        if (t[i] = '3') or (t[i] = '4') then  
            exit;  
        correct := true;  
    end;
```

Задача F. Тушенка

- Отношение 2:4:6 -- это то же самое, что 1:2:3
- Сделаем отношения такими, чтобы все числа в них были взаимно простыми.
- Для этого найдем НОД трёх чисел и разделим каждое из них на него.
- Чтобы можно было разделить тушенку между ребятами, нужно, чтобы количество банок тушенки делилось на сумму чисел в каждом полученном разбиении
- $\text{ans} \% (a+b+c) = \text{ans} \% (d+e+f) = 0$ (здесь a, b и c уже взаимнопростые и d, e и f тоже)
- Таким образом ответ -- наименьшее общее кратное(НОК) двух сумм.
- $\text{НОК}(a,b) = a * b / \text{НОД}(a,b)$

Задача G. <<Сапер>> с подсказками

- Обойдем все клетки поля. Для каждой клетки проверим, не нужно ли подсвечивать флажки вокруг нее.
- Для этого посчитаем количество флажков в клетках вокруг текущей. Если их больше, чем число, записанное в клетке, сохраним эти флажки в массив.
- Поскольку флажки нужно выводить без повторов, перед добавлением пройдем по массиву и проверим, нет ли в нем уже такого флажка.

Задача G. <<Сапер>> с подсказками

- Как обрабатывать соседние флажки:

```
procedure check_neighbour_flags(x, y: integer);
var
    new_x, new_y, dx, dy: integer;
begin
    for dx := -1 to 1 do
        for dy := -1 to 1 do
            begin
                new_x := x + dx;
                new_y := y + dy;
                if (new_x > 0) and (new_x <= N) and      // new_x внутри поля
                    (new_y > 0) and (new_y <= M) and      // new_y внутри поля
                    (field[new_x][new_y] = 'F')          // флажок ли здесь?
                then
                    //обработка соседнего флажка (new_x, new_y)
            end;
        end;
    end;
```

Задача Н. Асфальтоукладчик

- Максиму нужно как можно быстрее попасть к участку с остывшим асфальтом.
- Все такие участки находятся за асфальтоукладчиком, и, значит, в самой выгодной стратегии Максим идет ему навстречу.
- Как только Максим встретился с асфальтоукладчиком, он должен дойти до ближайшей полосы с остывшим асфальтом
- Это либо положенный вчера асфальт, либо положенный сегодня и уже остывший.

Задача Н. Асфальтоукладчик

- Если Максим пойдет по вчерашнему асфальту, то ему потребуется D/V_1 минут, чтобы до него дойти.
- Если Максим пойдет по свежему асфальту, то сначала ему потребуется $D/(V_1 + V_2)$ минут, чтобы дойти до асфальтоукладчика, а затем - $V_2 \cdot T/(V_1 + V_2)$ минут, чтобы дойти до застывшего асфальта (то есть оказаться на расстоянии $V_2 \cdot T$ от асфальтоукладчика).
- Еще надо перейти дорогу - это L/V_1 минут
- Итоговая формула:

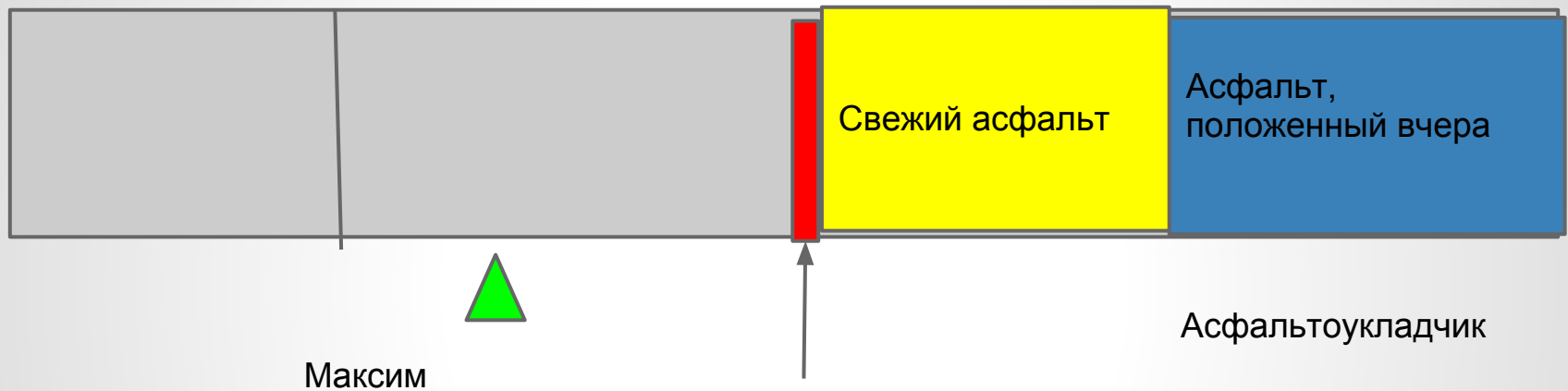
$$\min(D/V_1, (D + V_2 \cdot T)/(V_1 + V_2)) + L/V_1$$

Задача Н. Асфальтоукладчик

- Начальное состояние.



Задача Н. Асфальтоукладчик



Задача Н. Асфальтоукладчик

