



I (школьный) этап Всероссийской олимпиады школьников
2016/17 учебный год

Предмет	Класс	Дата	Время начала	Время окончания
информатика	9-11			

Методические указания к решению задач

Задача 1. «Арварохский хуб» (10 баллов)

Можно заметить, что двигаясь только вперед, хуб попадает в точки, координаты которых кратны 6. Количество ходов для перемещение в эти точки равно $(N \text{ div } 6)$. За один ход «вперед-назад» хуб перемещается на -1 позицию. Следовательно, в точку с произвольной координатой N можно попасть, выполнив $(N \text{ div } 6) + 1$ перемещение «вперед» и затем, выполняя перемещения «вперед-назад», количество которых равно $(6 - N \text{ mod } 6)*2$, попасть с требуемую точку. Таким образом, решение задачи сводится к простому линейному алгоритму. Например, на Паскале:

```
var n: longint;  
input, output: text;  
begin  
assign(input, 'input.txt');  
reset(input);  
readln(input, n);  
assign(output, 'output.txt');  
rewrite(output);  
if (n mod 6 = 0) then write (output, n div 6)  
else write (output, (n div 6 + 1) + (6 - (N mod 6)) * 2);  
close(output);  
close(input);  
end.
```

Задача 2. «Словарь иррашийского языка» (10 баллов)

Стандартная комбинаторная задача на подсчет количества размещений с повторениями. Известно, что

$A = N! / K!$. На небольших значениях переменных K и N будет работать программа циклического накопления произведения или с вызовом рекурсивной функции степени:

```
var k, m, n: integer;  
c: integer;  
function degr(x, y: integer): integer;  
begin  
if y=0 then degr:=1;  
if y=1 then degr:=x;  
if y>1 then degr:=x*degr(x, y-1);  
end;  
begin  
read(n, k, m);  
c:= degr(n, k);  
if (c mod m = 0) then c:=c div m else c:=(c div n) + 1;  
write(c);  
end.
```

И после нахождения количества размещений подсчитывает число страниц в словаре: как количество размещений, деленное на количество слов на странице, если число делится без остатка, иначе добавляет 1 для учета неполной страницы.

Задача 3. «Хумгат» (10 баллов)

Для решения задачи данные вводятся в двумерный массив размерности N. Параллельно выбирается самая левая точка (с минимальной координатой X). От нее будем строить ломаную. Далее перебираем



I (школьный) этап Всероссийской олимпиады школьников 2016/17 учебный год

Предмет	Класс	Дата	Время начала	Время окончания
информатика	9-11			

всевозможные ломаные, считаем их длину и выбираем минимальную длину. Расстояние между точками вычисляем по формуле: $S = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}$, полученные значения S округляем с избытком.

Задача 4. «Гонорар сыщика» (10 баллов)

Строка рассматривается как массив символов. Проходя по массиву, выбираем участки с символами, коды которых находятся в интервале от 48 до 57 (коды, соответствующие цифрам). Копируем данный участок в строку s и переводим в число. Затем продвигаемся далее по массиву и выбираем максимальной из возможных чисел, не превосходящее 1000000.

```
program z4;
var s,c: string;
x, max: longint;
i, j, n, kod: longint;
input, output: text;
begin
assign(input, 'input.txt');
reset(input);
readln(input, s);
x:=-1;
j:=1;
max:=-1;
while (j < length(s)) do
begin
i:=j;
while ((i <= length(s)) and (ord(s[i]) < 48) or (ord(s[i]) > 57)) do
i:=i+1;

n:=i;
if (n > length(s)) then break;

while ((i <= length(s)) and (ord(s[i]) = 48)) do
i:=i+1;
if (i-n > 0) then n:=i-1;

while ((i <= length(s)) and (ord(s[i]) >= 48) and (ord(s[i]) <= 57)) do
i:=i+1;
c:=copy(s, n, i-n);
if i-n <= 8 then val(c, x, kod);
if (x >= 0) and (x < 1000000) and (x > max) then max:=x;
j:=i;
x:=-1;
end;
assign(output, 'output.txt');
rewrite(output);
if max=-1 then writeln(output, 'NO') else writeln(output, max);
close(output);
end.
```

Задача 5. «Кумонская Заза» (10 баллов)

Прежде всего, нужно прочитать входные данные в двумерный массив.

Теперь составим схему решения задачи. Для каждой строки нам нужно однозначно определить цвет клеток, которые должны на ней оказаться. Этот цвет



**I (школьный) этап Всероссийской олимпиады школьников
2016/17 учебный год**

Предмет	Класс	Дата	Время начала	Время окончания
<i>информатика</i>	<i>9-11</i>			

(закодированный числами 1 или -1) будет храниться в массиве `color[i]`, который изначально заполнен нулями.

Для определения цвета клеток необходимо сделать следующее:

1) Проверить каждую горизонтальную костяшку.

- Если она состоит из двух квадратов разного цвета, то можно вывести NO и сразу завершить выполнение программы.
- Если в ряду имеются горизонтальные костяшки разного цвета, то тоже можно вывести NO и завершить работу программы.
- Если имеются горизонтальные костяшки только одного цвета, то переменной `color[i]` присвоить значение соответствующего цвета.

2) Просмотреть все вертикальные одноцветные костяшки и убедиться, что их цвет не противоречит цвету строки, а если цвет строки еще не определен, то определить его согласно расположению этих костяшек.

3) Рассмотреть вертикальные костяшки, состоящие из клеток разного цвета.

При этом возможны три ситуации:

- Цвета обеих строк, которые занимает такая костяшка, уже определены, — тогда достаточно убедиться, что эти цвета различны, а в противном случае вывести NO и завершить работу программы;
- Цвет первой строки, которую занимает такая костяшка, определен, а цвет второй — нет. В этом случае нужно определить цвет второй строки как противоположный цвету первой;
- Цвет первой строки (верхней из двух, которые занимает костяшка), не определен. Это означает, что в данной строке нет ни горизонтальных, ни одноцветных вертикальных костяшек, ни даже костяшек разного цвета, уходящих вверх (иначе бы мы определили цвет данной строки, рассматривая такую костяшку в предыдущей строке). Это также означает, что вся данная строка занята двухцветными вертикальными костяшками, уходящими вниз. И мы можем их перевернуть так, чтобы данная строка была одного цвета, а следующая — противоположного.

В случае же, если ни одна из вышеприведенных проверок не прервала программу с ответом NO, нам остается вывести ответ ZAZA.

Обратите внимание на некоторые технические детали реализации описанного выше алгоритма. Во-первых, в последней части мы использовали цикл `while` вместо `for`, чтобы иметь возможность пропускать строки, если в двух соседних строках все



I (школьный) этап Всероссийской олимпиады школьников
2016/17 учебный год

Предмет	Класс	Дата	Время начала	Время окончания
информатика	9-11			

костяшки двухцветные и вертикальные. Во-вторых, мы использовали условия вида $\text{if } \text{color}[i] * a[i,j] > 0$ вместо равносильного, но более длинного $\text{if } (\text{color}[i] > 0 \text{ and } a[i,j] > 0) \text{ or } (\text{color}[i] < 0 \text{ and } a[i,j] < 0)$, воспользовавшись тем, что произведение отрицательно только тогда, когда сомножители имеют разные знаки.

Пример решения на Паскале:

```
const N=8;
var color: array[1..N] of integer;
a: array[1..N,1..N] of integer;
i,j: integer;
flag: boolean;
input, output: text;
begin
  for i:=1 to N do color[i]:=0;
  assign(input, 'input.txt');
  reset(input);
  for i:=1 to N do
    for j:=1 to N do read(input,a[i,j]);
  close(input);
  flag:=true;
  //рассматриваем ряд на предмет наличия в нем горизонтально расположенных костяшек
  i:=1;
  while flag and (i<=N) do
    begin
      j:=1;
      while flag and (j<N) do
        begin
          if a[i,j]=-a[i,j+1] then flag:=false; //если костяшка двухцветная, то
          перестановка невозможна
          if (a[i,j]=a[i,j+1]) and (a[i,j]*color[i]<0) then flag:=false; //если
          костяшка одноцветная, но цвет не совпадает с цветом строки - перестановка невозможна
          if (a[i,j]=a[i,j+1]) and (color[i]=0) then color[i]:=a[i,j] div abs(a[i,j]);
          //если костяшки одноцветные, но цвет строки не определен, переопределяем его
          j:=j+1;
        end;
      i:=i+1;
    end;
  //рассматриваем вертикальные одноцветные костяшки
  i:=1;
  while flag and (i<N) do
    begin
      j:=1;
      while flag and (j<=N) do
        begin
          if a[i,j]=a[i+1,j] then
            begin
              if color[i]=0 then color[i]:=a[i,j] div abs(a[i,j]);
              if color[i+1]=0 then color[i+1]:=a[i+1,j] div
              abs(a[i+1,j]); //если костяшки одноцветные, но цвет строки не определен, переопределяем
              его
              if (a[i,j]*color[i]<0) or (a[i+1,j]*color[i+1]<0) then
                flag:=false; //если цвет костяшки не совпадает с цветом ряда, то перестановка
                невозможна
            end;
          j:=j+1;
        end;
      i:=i+2;
    end;
  //рассматриваем двухцветные вертикальные костяшки
  i:=1;
```



I (школьный) этап Всероссийской олимпиады школьников
2016/17 учебный год

Предмет	Класс	Дата	Время начала	Время окончания
<i>информатика</i>	<i>9-11</i>			

```
while flag and (i<N) do
begin
  for j:=1 to N do
    if (a[i,j]=-a[i+1,j]) then
      begin
        if(color[i]*color[i+1]=1) then flag:=false;
        if (color[i]<>0) and (color[i+1]=0) then color[i+1]:=-color[i];
        if (color[i+1]<>0) and (color[i]=0) then color[i]:=-color[i+1];
        end;
      i:=i+2;
    end;
  assign(output, 'output.txt');
  rewrite(output);
  if flag then writeln(output, 'ZAZA') else writeln(output, 'NO');
  close(output);
  end.
```