# Ответы на задания школьного этапа всероссийской олимпиады школьников

# по информатике и ИКТ 2017/2018 учебный год

# 

# 10 класс (максимум – 100 баллов)

1. **Вот так забег! (15 баллов)**

Ответ: пятым финишировал бегун под номером 1147.

**РЕШЕНИЕ.** Проанализируем номера финишировавших бегунов: 12219, 14176, 7133, 990. Данная числовая последовательность, на первый взгляд, не упорядочена, т.к. после первого числа сначала следует большее, а потом – меньшее. Но, обращая внимание на три последних цифры в числах и, выписав их, замечаем, что прослеживается довольно простая зависимость: 219, 176, 133, т.е. каждое последующее число меньше предыдущего на 43. Действительно,

219-176=43; 176-133=43. Проверяем догадку: 133-90=43. Следовательно, следующее число 90-43=47. Что же представляют из себя старшие разряды номеров финишировавших спортсменов? Можно заметить, что они представляют из себя сумму цифр последующих разрядов, т.е.: 2+1+9=12; 1+7+6=14; 1+3+3=7;

9+0=9. Значит, необходимо найти сумму цифр полученного нами числа 47.

4+7=11. Следовательно, номер спортсмена, финишировавшего пятым, равен 1147.

**Критерии оценки.**

В работе дан только правильный ответ без анализа и объяснений – 0 баллов.

В работе обнаружена закономерность между числами, определён шаг арифметической прогрессии, но правильный ответ не получен – 5 баллов.

В работе правильно определён шаг арифметической прогрессии, обнаружена закономерность между числами и суммами их цифр, но правильный ответ не сформулирован – 10 баллов

В работе правильно определён шаг арифметической прогрессии, обнаружена закономерность между числами и суммами их цифр, правильный ответ сформулирован и записан – 15 баллов.

1. **Своя ноша не тянет. (10 баллов)**

Ответ: козёл всегда перевозит масло.

**РЕШЕНИЕ.** Составим таблицу всех вариантов поклажи на животных:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № варианта | **А** | **Б** | **В** | **Г** | **Д** | **Е** | **Ж** | **З** |
| **Осёл** | М | М | М | М | Ф | Ф | Ф | Ф |
| **Козёл** | М | М | Ф | Ф | М | М | Ф | Ф |
| **Верблюд** | М | Ф | М | Ф | М | Ф | М | Ф |

По первому условию, если осёл перевозит финики, то козёл – масло. Это исключает варианты Ж и З.

По второму условию, если осёл перевозит масло, то верблюд – финики. Это исключает варианты А и В.

По третьему условию, если козёл перевозит финики, то верблюд – масло. Это исключает вариант Г.

|  |  |  |  |
| --- | --- | --- | --- |
| № варианта | **Б** | **Д** | **Е** |
| **Осёл** | М | Ф | Ф |
| **Козёл** | М | М | М |
| **Верблюд** | Ф | М | Ф |

Проверяем оставшиеся варианты Б, Д, Е на наличие противоречий.

В варианте Б нет противоречий: и осёл, и козёл перевозят масло, а верблюд – финики. Это соответствует условию второму, где говорится: если осёл перевозит масло, то верблюд – финики. Условие первое сообщает: если на осле доставляют финики, то на козле – масло. Однако, если осёл не перевозит финики, то козёл должен перевозить либо финики, либо масло. Из условия третьего следует: если козёл тащит на себе финики, то верблюд – масло. Но если козёл не перевозит финики, то тогда верблюд может перевозить и финики, и масло.

В варианте Д нет противоречий: на осле доставляют финики, а на козле и верблюде – масло. По условию первому, если осёл тащит на себе финики, то козёл – масло. Раз осёл не перевозит масло, то, учитывая условие второе, верблюд может перевозить и масло, и финики. Раз козёл не перевозит финики, делаем вывод из условия третьего, что верблюд может перевозить и финики, и масло.

В варианте Е нет противоречий: и осёл, и верблюд перевозят финики, а козёл- масло. В условии первом говорится: если на осле доставляют финики, то на козле - масло. Т.к. осёл перевозит финики, а не масло, то делаем вывод из условия второго, что верблюд перевозит и то, и другое. Поскольку козёл перевозит масло, а не финики, точно так же можно предположить (исходя из условия третьего), что нет никакого противоречия в том, что верблюд перевозит финики.

Единственное животное, чья поклажа теперь известна наверняка, - козёл. Во всех трёх возможных ситуациях (Б,Д,Е) он перевозит только масло.

**Критерии оценки.**

В работе дан только правильный ответ без объяснений – 0 баллов.

В работе имеются правильные рассуждения, но работа не доделана до конца, не получен правильный ответ – 5 балл.

В работе проанализированы все возможные случаи поклажи животных, получен правильный ответ – 10 баллов.

1. **Что за алгоритм? (15 баллов)**

Ответ: данный алгоритм выполняет функцию нахождения наибольшего общего делителя двух натуральных чисел (алгоритм Евклида).

**РЕШЕНИЕ.**

|  |  |
| --- | --- |
| N команды | Команда |
| 2 | НАЧАЛО |
| 5 | ВВОД М,N |
| 7 | ПОКА М≠N, ПОВТОРЯЙ |
| 4 | ЕСЛИ М>N, ТО |
| 6 | М:=М-N |
| 11 | ИНАЧЕ |
| 10 | N:=N-M |
| 9 | КОНЕЦ ВЕТВЛЕНИЯ |
| 8 | КОНЕЦ ЦИКЛА |
| 3 | ВЫВОД М |
| 1 | КОНЕЦ |

Для доказательства трассируем алгоритм:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Шаг** | **Операция** | **М** | **N** | **Условие** |
| 1 | ВВОД М | 20 |  |  |
| 2 | ВВОД N |  | 10 |  |
| 3 | М≠N |  |  | 20≠10, ДА |
| 4 | М>N |  |  | 20 > 10, ДА |
| 5 | М:=М-N | 10 |  |  |
| 6 | М≠N |  |  | 10≠10, НЕТ |
| 7 | ВЫВОД М | 10 |  |  |

Таким образом, трассировка алгоритма показывает, что он находит и выводит наибольший общий делитель двух натуральных чисел М и N.

**Критерии оценки.**

В работе дан только правильный ответ без объяснений – 0 баллов.

В работе более 50% команд поставлены правильно, но работа не доделана до конца, правильный ответ не получен – 3 баллов.

В работе все команды алгоритма стоят на своих местах, но нет трассировки алгоритма, не получен правильный ответ – 5 баллов.

В работе все команды алгоритма стоят на своих местах, но нет трассировки алгоритма, получен правильный ответ – 10 баллов.

В работе все команды алгоритма стоят на своих местах, произведена трассировка алгоритма, получен правильный ответ – 15 баллов.

1. **«Digit»** **(25 баллов)**

Решение на языке Паскаль:  
program z1;  
var x,a,b,c,n:integer; f,f1:text;  
begin  
write(' vvedite chislo ');  
readln(x);  
a:=x div 100;  
b:=x mod 100 div 10;  
c:=x mod 10;  
if (a < b) and (a < c) then N:=10\*b+c

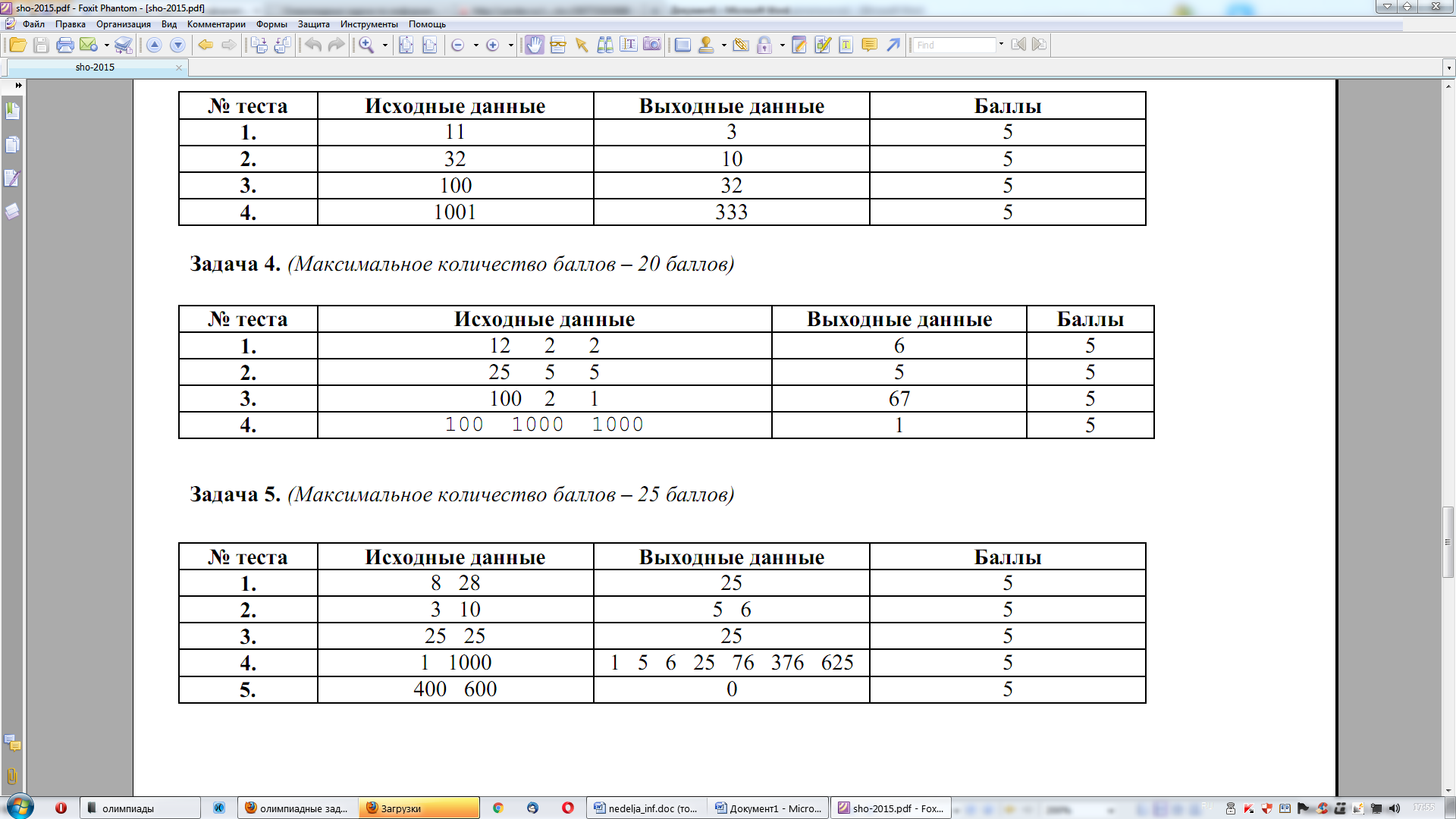
else

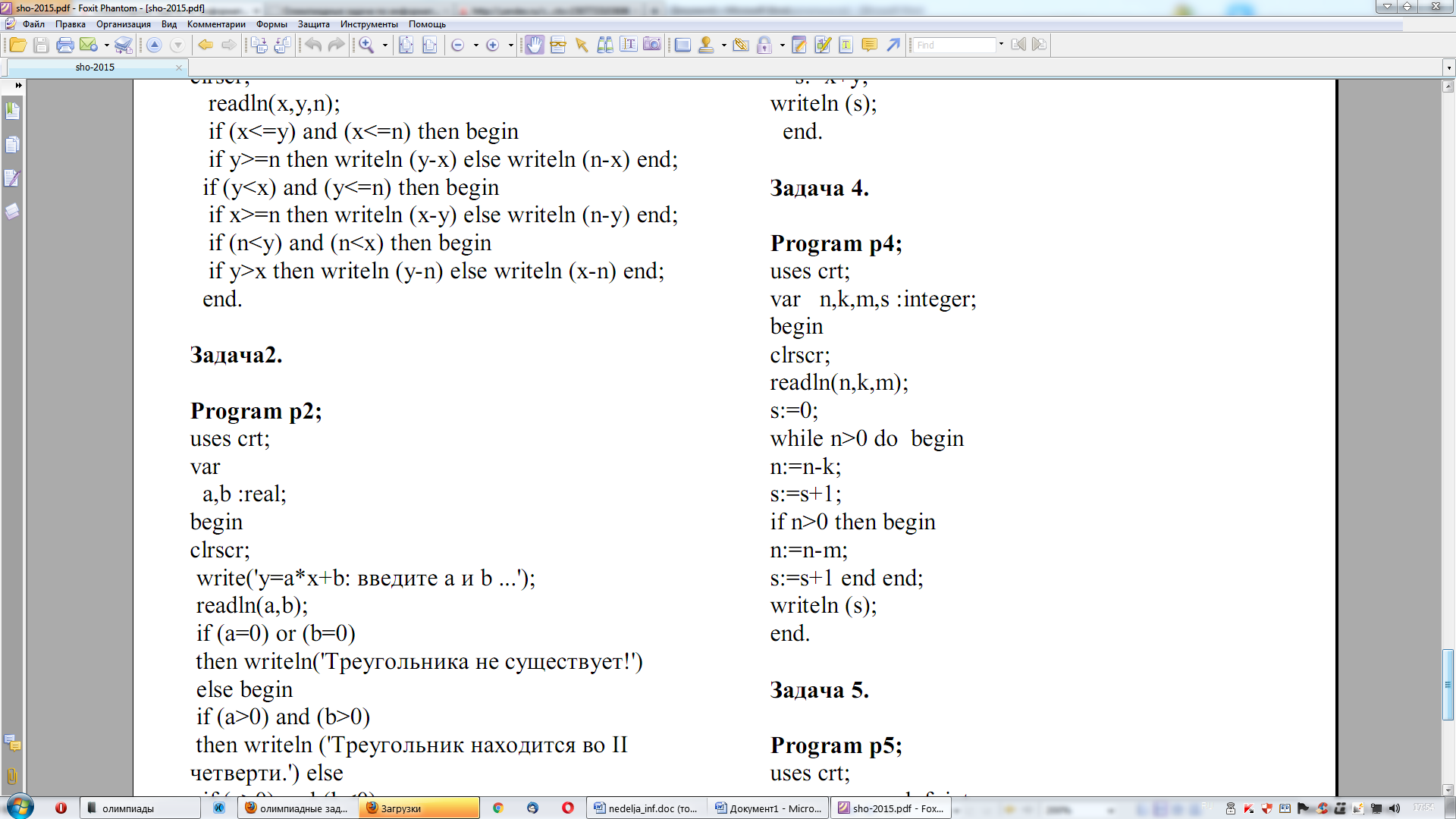
if (a > b) and (b < c) then N:=10\*a+c

else N:=10\*a+b;  
writeln('N=',N);

Readln;  
end.

1. **«Пассажиры» (35 баллов)**





# 11 класс (максимум – 100 баллов)

1. «Заклинания» (5 баллов)

Ответ 125

1. **«Неизвестное» (10 баллов)**

Ответ 32

1. **«Перевозки»** **(20 баллов)**

Обозначим: k - номер рейса судна, i - номер очередного груза, s - масса груза на судне в k-том рейсе. Решать задачу будем так: если на судно в k-том рейсе можно поместить ещё один груз, то мы грузим его и берём следующий, если груз не может быть размещен, то перевозим его следующим рейсом (увеличиваем k).   
Основная часть соответствующей программы будет иметь вид:

k:=1;

i:=1;

s:=0;

repeat

if s+m[i]<=50 then

begin

s:=s+m[i];

 i:=i+1;

end

else begin

k:=k+1;

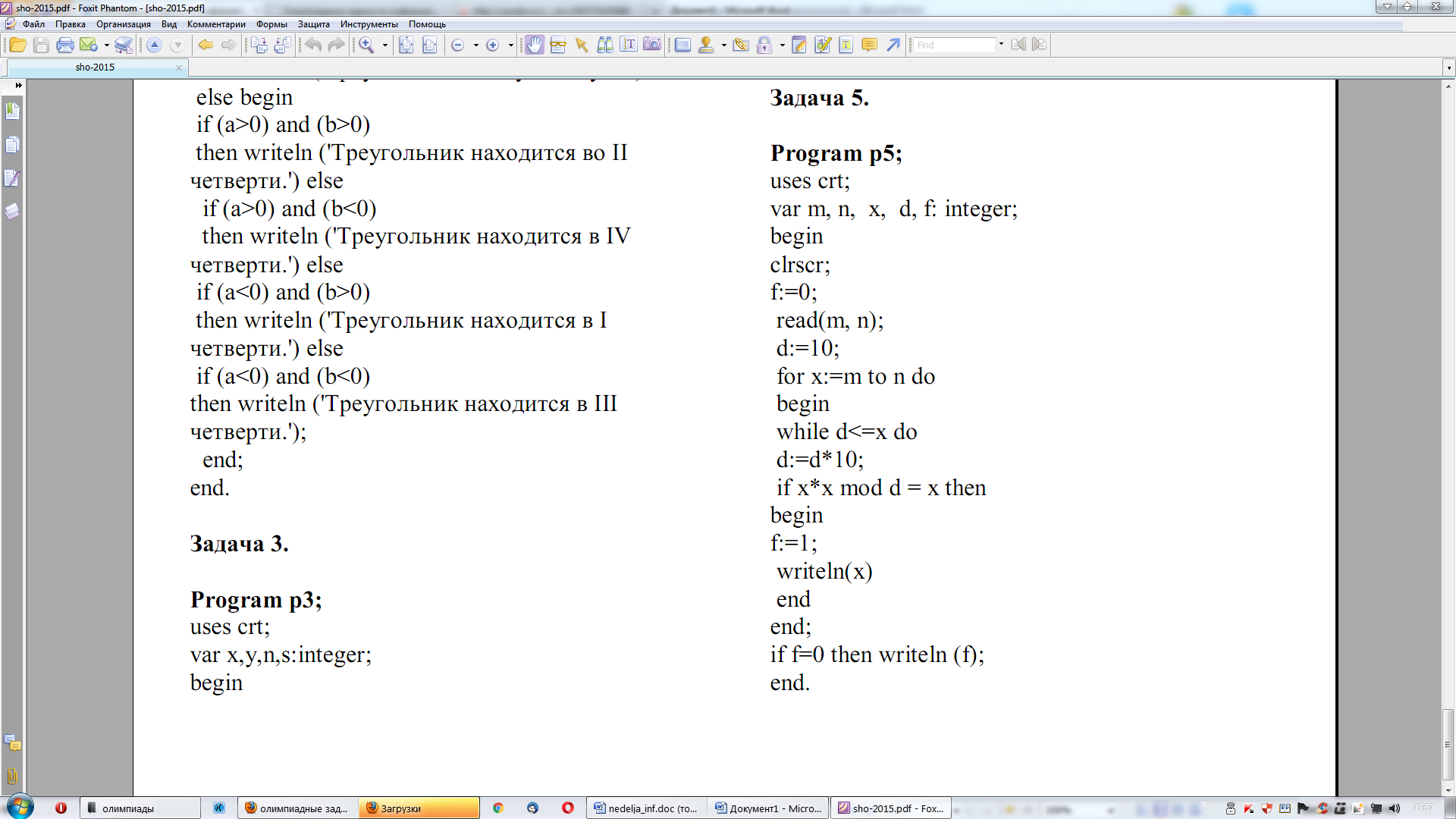
s:=0;

end;

until i>15;

writeln('Всего потребовалось', k,' рейсов');

1. **«Автоморфные числа» (25 баллов)**

****

|  |  |
| --- | --- |
| 1 | 1 |
| 5 | 25 |
| 6 | 36 |
| 25 | 625 |
| 76 | 5776 |
| 376 | 141376 |
| 625 | 390625 |

Выполнив приведенную программу, ЭВМ нашла в интервале [1, 1000] следующие автоморфные числа (в первом столбце напечатаны автоморфные числа, во втором - их квадраты):

1. **«Острова» (40 баллов)**

Итак, это классическая задача на поиск в глубину графа. Понятно, что надо обходить матрицу и каким-то образом вычислять количество островов. Вариант решения такой: после того, как мы попадаем на остров, надо это зафиксировать увеличив переменную-результат на единицу. Чтобы второй раз не посчитать один и тот же остров, сразу после посещения необходимо его уничтожить, т.е. присвоить всем клеткам острова значение ноль.

Поскольку тест задачи не слишком мал, стоит написать процедуру уничтожения островов, назовем ее "count". Чтобы во время выполнения процедуру не "выскочить" за пределы массива, сделаем его не размером N x N, а размеров N+2 x N+2, это даст  нам возможность окружить искомый массив размером N x N  нулями.

**const**

m = 102;

**var**

a:array [1..m, 1..m] of integer;

i,j,n,res: integer;

input,output: text;

**procedure** count(i,j: integer);

**begin**

**if** a[i,j] <> 1 **then**

**exit;**

a[i,j] := 0;

count(i + 1, j);

count(i - 1, j);

count(i, j + 1);

count(i, j - 1);

**end;**

**begin**

res:= 0;

assign(input,'input.txt');

reset(input);

assign(output,'output.txt');

rewrite(output);

read(input, n);

{Заполняем массив нулями}

**for** i:= 1 **to** n+2 **do**

**for** j:=1 **to** n+2 **do**

a[i,j]:=0;

{Считываем матрицу из файла}

**for** i:= 2 **to** n+1 **do**

**for** j:=2 **to** n+1 **do**

read(input,a[i,j]);

{Обходим матрицу в поиске островов}

**for** i := 2 **to** n+1 **do**

**for** j := 2 **to** n+1 **do**

**if** a[i,j] = 1 **then**

**begin**

inc(res);

count(i,j);

**end;**

write(output,res);

close(input);

close(output);

**end.**