

**ФГОБУ ВО «Финансовый университет при Правительстве  
Российской Федерации»  
ВСЕРОССИЙСКАЯ ОЛИМПИАДА ПО ИНФОРМАТИКЕ  
«МИССИЯ ВЫПОЛНИМА. ТВОЕ ПРИЗВАНИЕ – ФИНАНСИСТ!»  
ОЧНЫЙ ЭТАП, 2021 год**

**Инструкция участнику олимпиады**

Продолжительность олимпиады – 240 минут (4 астрономических часа).  
Олимпиадное задание состоит из пяти задач. Для каждой задачи указан ее вес в баллах.

Участник олимпиады самостоятельно определяет последовательность выполнения задач. На одном из языков программирования – C/C++, C#, Visual Basic, Pascal или Python – разработайте *консольные* программы для решения перечисленных ниже задач.

При выполнении задания участник формирует каталог в имени которого указывает свое ФИО. В данном каталоге формирует пять каталогов: Task1; Task2; Task3; Task4; Task5. Решение задачи размещаются в каталоге с соответствующим номером (см. рис П.1)

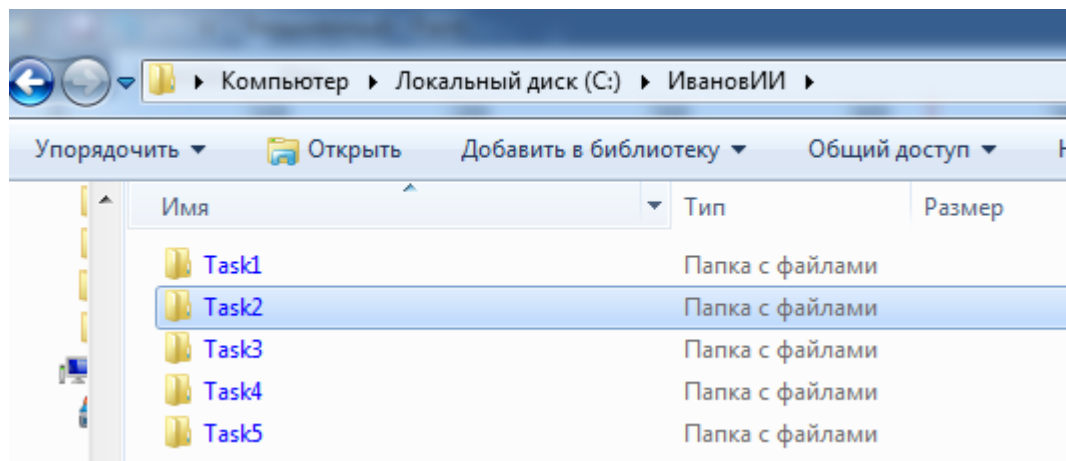


Рисунок П.1 – Структура каталога участника олимпиады

Участник олимпиады должен предоставить членам комиссии на проверку только файлы с исходными текстами программ, которые должны быть названы участником олимпиады в соответствии с выполняемым заданием, например, для языка Python: Task1.py.

Расширение файла должно соответствовать языку. Переименуйте файлы перед сдачей работы, если это необходимо. (см пример на рис. П.2)

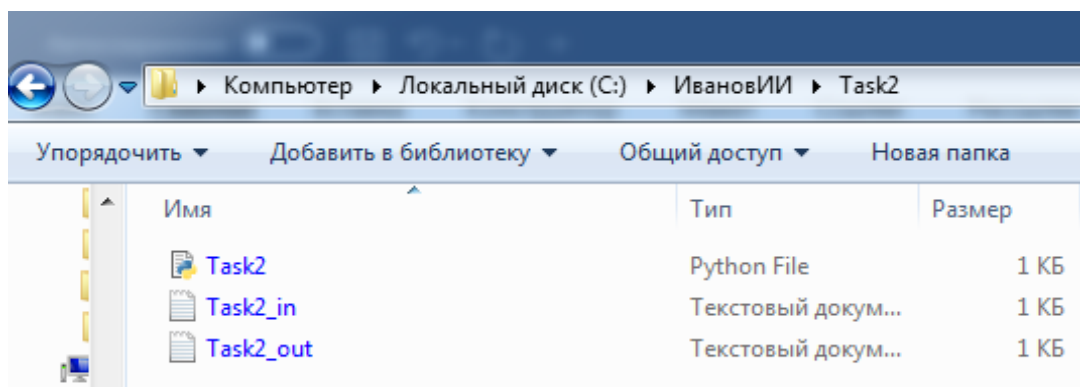


Рисунок П.2 – Размещение файлов в рамках папки задачи

В начале каждой программы должен находиться комментарий с ФИО участника, вариант, номером задачи, языком программирования, средой программирования.

Например, для C-подобных языков: // Иванов И. И., вариант 1, задача 1, Python 3.7.3, Spyder 3.3.6.

Если файлы с решением задачи, исходных и результирующих данных имеют некорректные названия и/или отсутствует первая строка комментариев, и/или размещены в каталоге участника без учета требований к структуре, то члены комиссии данное решение не оценивают и баллы за решение задания не начисляются.

При решении задач в качестве файлов с исходными данными и выходными данными используется только текстовый файл с расширением \*.txt. Если в задаче программной реализации используется файлы с исходными данными и/или выходными данными, то кроме файла с исходным текстом требуется выслать соответствующие файлы. Например, для задания 2 требуется использовать исходные данные из файла, тогда название файла должно быть Task2\_in.txt. если в задании 2 требуется сформировать текстовый файл с результатами исполнения программы, то название файла должно быть Task2\_out.txt. Число после слова Task соответствует номеру решенного задания, «\_in» определяет, что файл с исходными данными, «\_out» определяет, что в файле хранятся результаты исполнения кода над исходными данными. Все текстовые файлы с исходными данными создаются участником самостоятельно, в соответствии с представленными в задачах примерами и шаблонами.

По окончании работы над заданиями участник формирует архив с содержимым решений в соответствии с предложенной выше структурой и размещает его в информационной системе Финансового Университета при Правительстве РФ. Расширение архивного файла должно быть rar или zip (пример см. рис П.3-П.5).

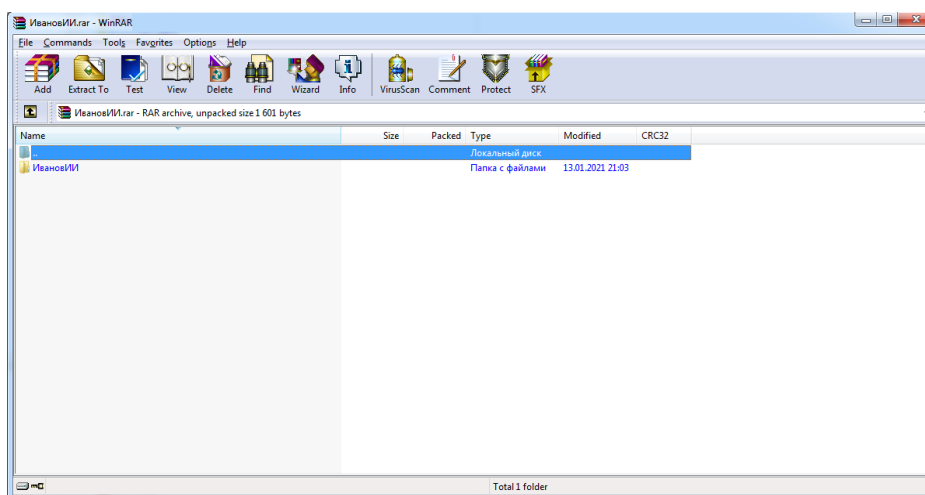


Рисунок П.3 – Пример первого уровня содержимого архива

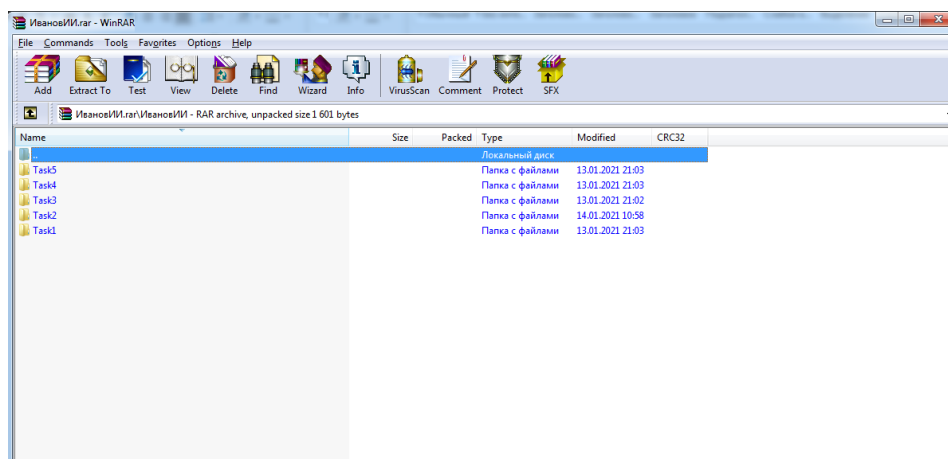


Рисунок П.3 – Пример второго уровня содержимого архива

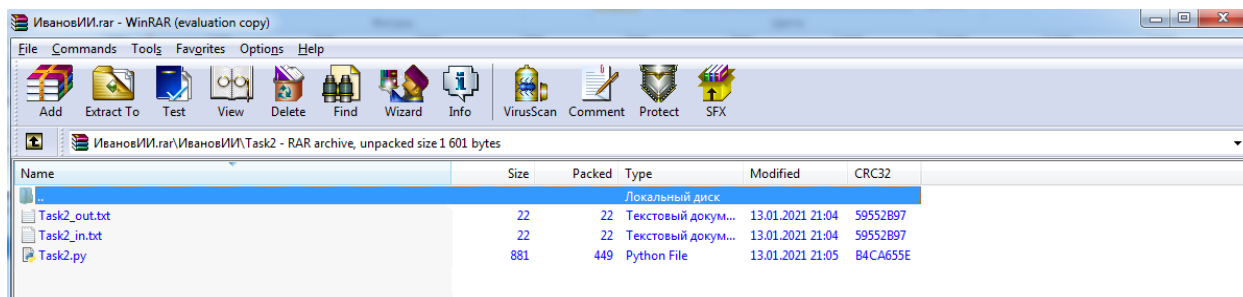


Рисунок П.3 – Пример третьего уровня содержимого архива

Члены комиссии, при проверке заданий, используют операционные системы семейства Windows (версии 7-10). Если члену комиссии не удастся запустить файл с исходным кодом на исполнение (например, программа не выполняется в перечисленных ОС и/или не находит файл с исходными данными, и/или не формирует результирующий файл и т. д.), то задание считается не выполненным.

Максимальное количество баллов, которые может набрать участник – 100.

При оценивании решения задачи члены жюри могут снизить баллы за следующие недостатки:

- неполное соответствие решения условию;
- применение неэффективного алгоритма;
- решение задачи только для частного случая;
- отсутствие проверок, приводящих к снижению надежности программы;
- низкое качество интерфейса пользователя;
- несоответствие решения пулу тестовых значений;
- плохая читабельность текста программы и т. д.

При обнаружении использования участником: посторонней помощи в любом проявлении, средств интернет, мобильных устройств и других приемо-передающих устройств, способствующих решению заданий олимпиады, не используя собственные знания, приведут к исключению участника и аннулированию его результатов.

## Олимпиадные задания по информатике Вариант 1

### Задача 1 (8 баллов)

Ване стало интересно, существуют ли такие пятизначные числа, которые делятся на 124 без остатка и при этом сумма цифр, соответствующих разрядам тысяч, сотен и десятков в 6200 раз меньше самого числа. Вы - начинающий программист и решили помочь Ване с данной задачей. Напишите программу, выводящую все числа, удовлетворяющие критериям, на экран. Если таких чисел нет, то выведите на экран сообщение «No».

### Задача 2 (12 баллов)

Требуется реализовать программное средство для расчета периметра и площади треугольника. Исходные данные, в виде длины сторон треугольника, программа берет из созданного текстового файла с именем Task2\_in.txt (создать самостоятельно с помощью Блокнота в соответствии с примером на рисунке 2.1). Значения длины сторон должны быть разделены «;», целая часть числа отделяется от дробной «.», конец строки обозначается «!».

Программная реализация должна включать функцию, которая принимает в качестве параметров три стороны треугольника и вычисляет его площадь и периметр (округление до десятых). Полученные результаты вычисления должны быть записаны в файл Task2\_out.txt (пример см. рис. 2.2). Если треугольник с заданными сторонами не

существует, то программа должна вернуть строку «No» в текстовый файл Task2\_out.txt (пример файла Task2\_out.txt для несуществующего треугольника см. рис. 2.3).

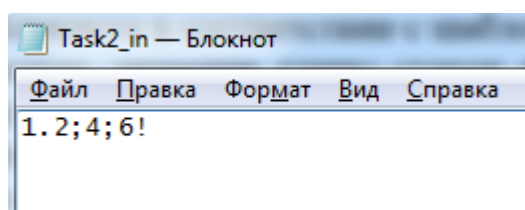


Рисунок 2.1 – Пример текстового файла Task2\_in.txt с исходными значениями, соответствующего шаблону

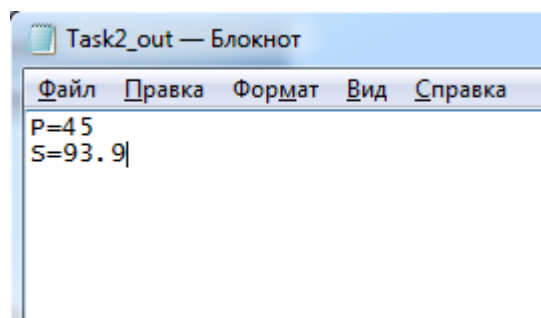


Рисунок 2.2 – Пример текстового файла Task2\_out.txt для существующего треугольника

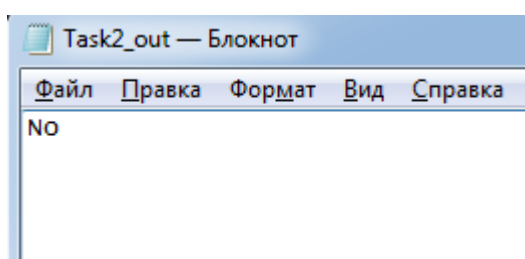


Рисунок 2.3 – Пример текстового файла Task2\_out.txt для несуществующего треугольника

### Задача 3 (15 баллов)

Выполнить программную реализацию расчета значений логической функции четырех переменных (см. рис. 3.1) и вывод результата в файл Task3\_out.txt.

$$F = (\text{not } A \text{ and } C \text{ or } B \text{ or not } D) \text{ and } (B \text{ and } C \equiv C \text{ and } D) \text{ and } (\text{not } B \text{ and } C \rightarrow A)$$

Рисунок 3.1 – Логическая функция четырех переменных

В функции обозначения соответствуют:

- *or* – логическое ИЛИ (дизъюнкция);
- *and* – логическое И (конъюнкция);
- *not* – логическое отрицание (НЕ);
- $\rightarrow$  – импликация;
- $\equiv$  – эквивалентность.

Исходные данные требуется считать из файла Task3\_in.txt строго в указанном порядке. (пример см. рис. 3.2)

```

Файл  Правка  Формат  Вид  Справка
ABCD
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111|

```

Рисунок 3.2 – Файл Task3\_in.txt

Значение F вычисляется разработанной программой (за ручное решение с записью в файл Task3\_out.txt баллы не начисляются), и записывается в файл Task3\_out.txt.

На рисунках 3.3 а) и 3.3 б) показаны примеры файлов Task3\_in.txt и Task3\_out.txt, заполненных в соответствии с требованиями, для логической функции от двух переменных  $F = A \text{ and } B$ .

<pre> Файл  Правка  Формат  Вид  Справка AB 00 01 10 11  </pre>	<pre> Файл  Правка  Формат  Вид  Справка ^ F 0 0 0 1  </pre>
---	--

а)

б)

Рисунок 3.3 – а) Файл Task3\_in.txt, б) Файл Task3\_out.txt

#### Задача 4 (30 баллов)

Бизнесмен собирается в путешествие по миру. Он знает дату начала своего путешествия и знает, сколько времени он проведёт путешествуя. Время путешествия представляет собой количество часов и минут, которое пройдёт от момента прибытия бизнесмена в аэропорт для вылета из своего города до момента его выхода из аэропорта в своем городе. Однако, как это обычно бывает из-за сложности оперирования с датами, бизнесмену сложно высчитать, когда он вернётся домой. При расчете следует учитывать, что дорога от его дома до аэропорта в среднем занимает 2 часа 32 минуты.

В программной реализации необходимо рассчитать дату и время возвращения бизнесмена домой.

Исходные данные программа должна брать из созданного текстового файла с именем Task4\_in.txt (создать самостоятельно с помощью Блокнота в соответствии с примером на рисунке 4.1). В котором содержится следующая информация:

- dd.mm.YYYY h:min, где dd – день, mm – месяц, YYYY – год, h – часы, min – минуты выхода бизнесмена из дома;
- h:min, где h – часы, min – минуты длительности путешествия, без учета времени, которое потратит бизнесмен на дорогу в аэропорт и обратно.

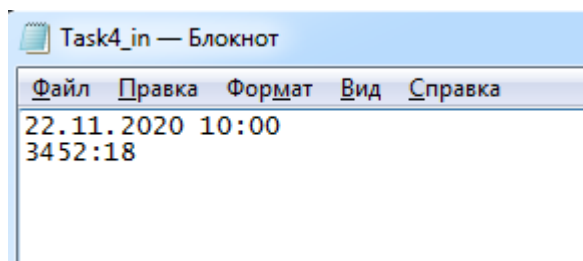


Рисунок 4.1 – Пример файла Task4\_in.txt, оформленный в соответствии с требованиями

Ответ выводится в файл Task4\_out.txt, в соответствии с шаблоном (см. рис. 4.2 )

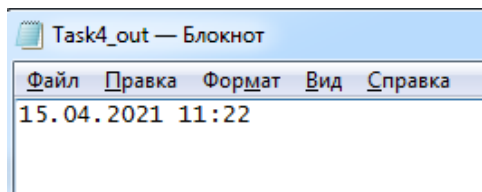


Рисунок 4.2 – Пример файла Task4\_out.txt, оформленный в соответствии с требованиями

При использовании стандартных библиотек для работы с датой и временем снимается 15 баллов.

### Задача 5 (35 баллов)

Испанские шашки один из вариантов игры в шашки. Распространены в Испании, Португалии, во многих странах Латинской Америки, на Филиппинах, в Скандинавии и в некоторых районах Германии.

В задаче требуется разработать программу, которая позволяет вычислить ход, который должен совершить игрок в соответствии с правилами испанских шашек (см. правила ниже).

Для программной реализации алгоритма вычисления хода поле для игры представляется текстовым файлом. На рисунке 5.1 показано кодирование игрового поля:

- «+» - черное поле;
- «\*» - белое поле.

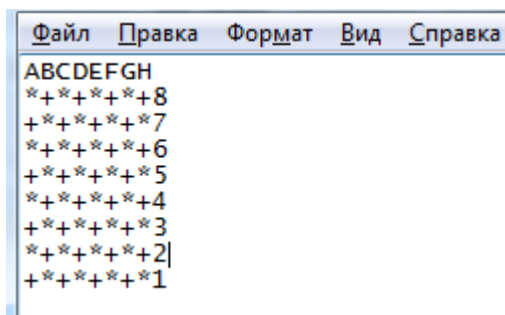


Рисунок 5.1 – Кодирование игрового поля.

В шаблоне кодирования шашек на поле используются следующие обозначения:

- обычная белая шашка изображается символом «l»;
- дамка белого цвета изображается «L»;
- обычная черная шашка изображается символом «b»;
- дамка черного цвета «B».

Пример закодированного расположения шашек на доске с помощью текстового файла Task5\_in.txt смотрите на рисунке 5.2

```

Файл  Правка  Формат  Вид  Сп
ABCDEFGH
*+*+*+*+8
+*+*+*+*7
*+*+b+b+6
+*+*+*+*5
*+b+*+*+4
+1+*+L+B3
*+*+*+1+2
+1+*+*+*1|

```

Рисунок 5.2 – Пример расположения шашек на игровом поле в файле Task5\_in.txt

Запись вариантов хода осуществляется в соответствии со следующими правилами:

- для записи хода шашки обозначают сначала поле, на котором шашка стояла, затем ставят тире и записывают поле, на которое она ставится (пример a2-b3).
- при записи взятия (боя) вместо тире ставится двоеточие (пример b3:d5);
- при взятии одним ходом нескольких шашек запись хода производится следующим образом: сначала записывается поле, с которого шашка начала свой ход, затем ставится двоеточие и обозначается поле, на которое она встала после боя (пример b3:f7).
- если необходимо отметить направление взятия, то после записи поля, с которого начался бой, последовательно записывают обозначение полей, на которых совершалось изменение направления. Между обозначениями отдельных полей ставится двоеточие (пример f3:d5:f7:h5).

Задача программной реализации состоит в следующем:

- из заданного расположения шашек в файле Task5\_in.txt (пример см. рис. 5.3), требуется найти ход в соответствии с правилами испанских шашек;

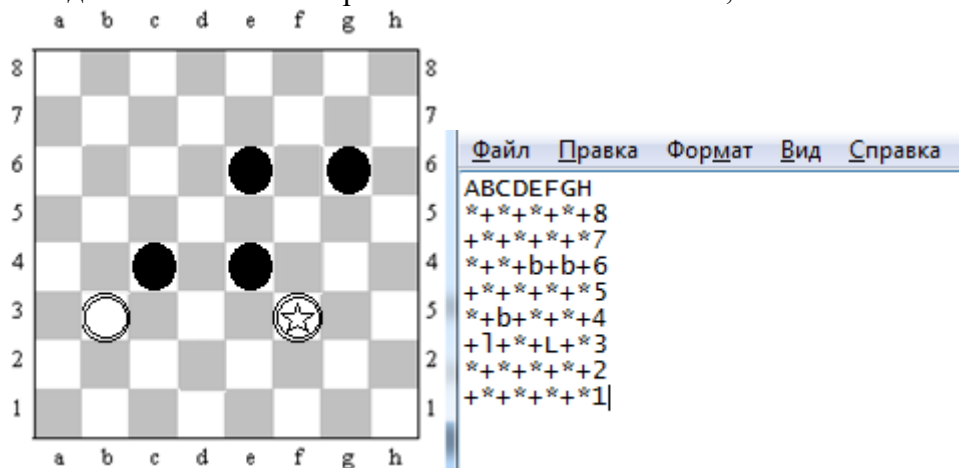


Рисунок 5.3 - Пример отображения шашек на доске и в файле Task5\_in.txt (шашка со звездочкой - дамка)

- право хода принадлежит белым шашкам;
- ход требуется записать в файл Task5\_out.txt (см. рис. 5.4);

```

Файл  Правка  Формат  Вид  Справка
f3:d5:f7|h5

```

Рисунок 5.4 – Пример выходного файла Task5\_out.txt (право хода у белых шашек)



- если количество ходов более одного, и они могут быть осуществлены, в соответствии с правилами испанских шашек, т. е. у игрока есть выбор (см. рис 5.5 и рис. 5.6), то ходы записываются в столбик.
- если ход невозможен, то требуется осуществить запись «No» в файл Task5\_out.txt.

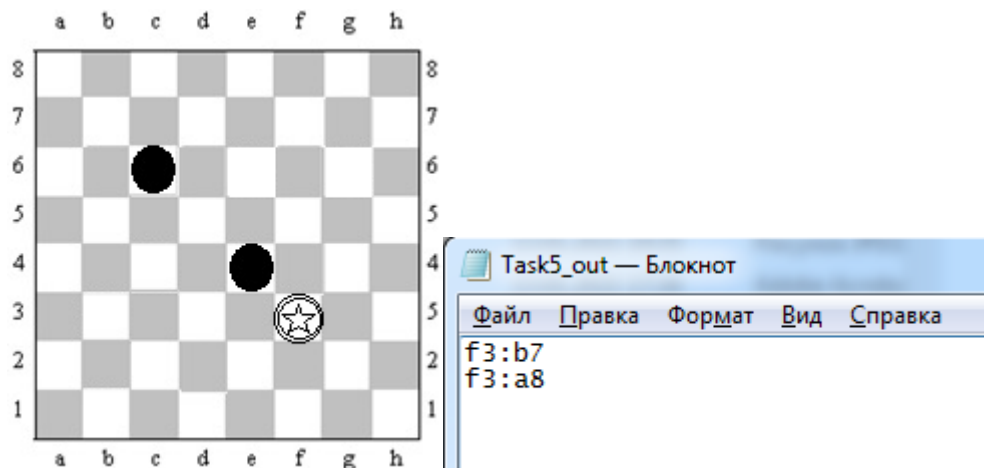


Рисунок 5.5 – Пример более чем одного варианта хода (право хода у белых шашек)

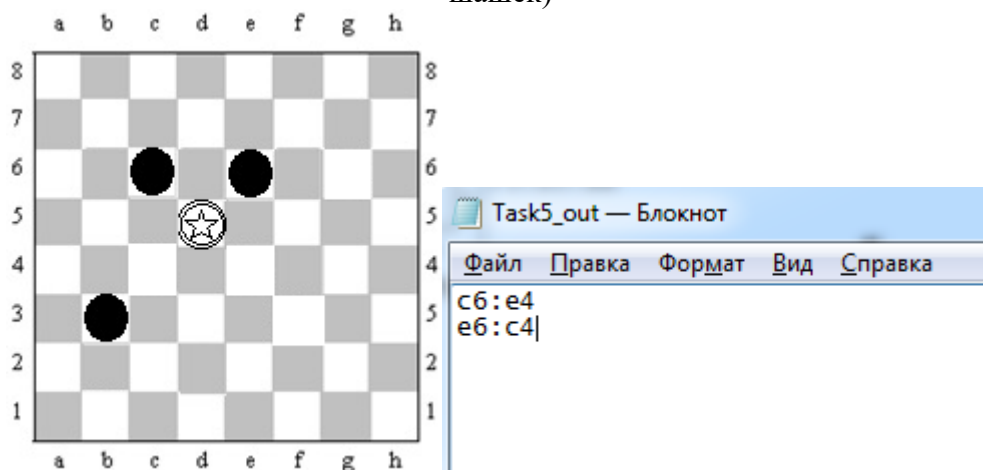


Рисунок 5.6 – Пример более чем одного варианта хода (право хода у черных шашек)

### Правила игры в испанские шашки:

Цель игры - уничтожить все шашки противника или лишить их возможности хода («запереть»). Игра проводится на доске 8×8 клеток. Доска располагается между партнерами таким образом, чтобы слева от играющего находилось тёмное угловое поле. В начальной позиции у каждого игрока по 12 шашек, расположенных в первых трёх рядах на белых клетках.

Первый ход делают белые шашки. «Простые» шашки могут ходить по диагонали на одно поле вперёд и бить только вперёд. Назад простые шашки бить не могут!

При достижении последнего (восьмого от себя) горизонтального ряда простая шашка превращается в дамку. Дамка ходит на любое количество клеток вперёд и назад. Взятие шашки соперника является обязательным.

При нескольких вариантах взятия бить нужно максимально возможное количество шашек соперника, а при равных вариантах боя нужно бить максимальное количество дамочек.

Дамка бьет по диагонали, как вперед, так и назад и становится на любое свободное поле после побитой шашки.



Если простая шашка достигла последнего ряда во время взятия, то она превращается в дамку и останавливается, даже при возможности продолжить взятие.