

**ФГОБУ ВО «Финансовый университет при Правительстве
Российской Федерации»
ВСЕРОССИЙСКАЯ ОЛИМПИАДА ПО ИНФОРМАТИКЕ
«МИССИЯ ВЫПОЛНИМА. ТВОЕ ПРИЗВАНИЕ – ФИНАНСИСТ!»
ОЧНЫЙ ЭТАП, 2022 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)

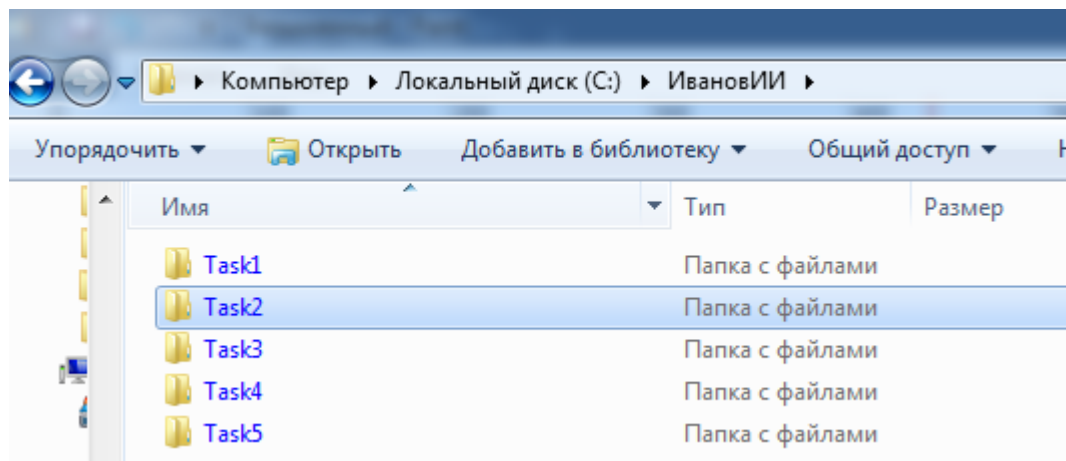


Рисунок П.1 – Структура каталога участника олимпиады

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)

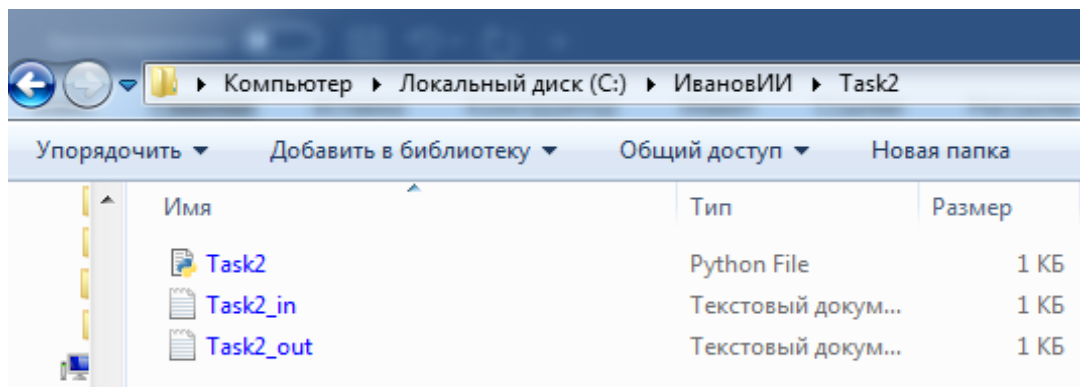


Рисунок П.2 – Размещение файлов в рамках папки задачи

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

Например, для С-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 программы требуется выслать соответствующие файлы. Например, для задания 2 требуется использовать исходные данные из файла, тогда название файла должно быть в соответствии с заданием (например, в задании 2 Task2_in.txt), если в задании требуется сформировать выходной текстовый файл, то его название также должно соответствовать указанному в задании (например, в задании 2 Task2_out.txt). 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).

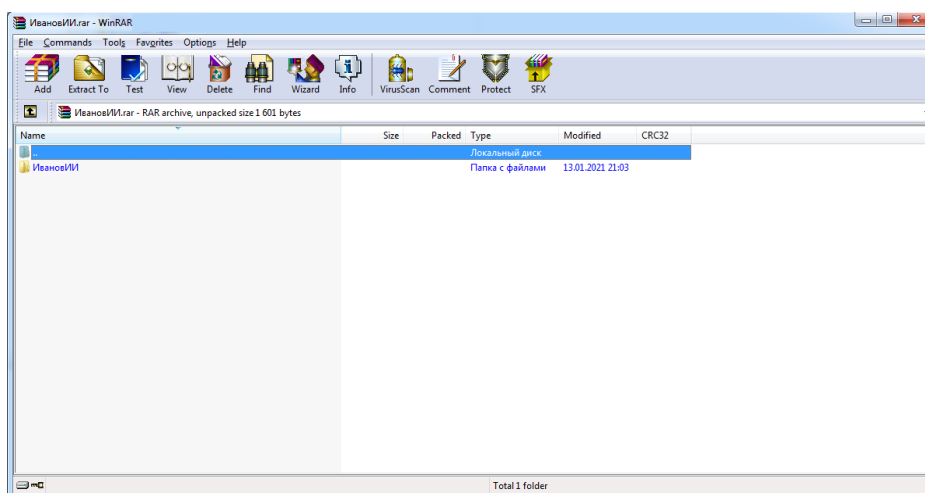


Рисунок П.3 – Пример первого уровня содержимого архива

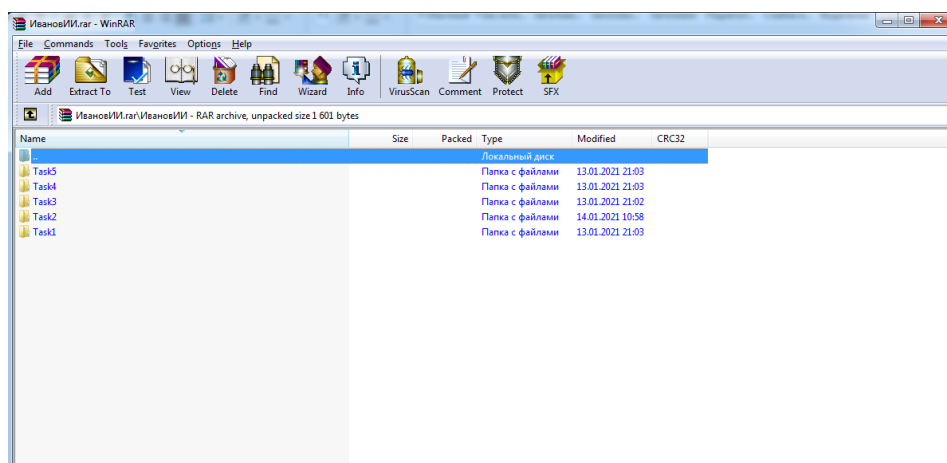


Рисунок П.3 – Пример второго уровня содержимого архива

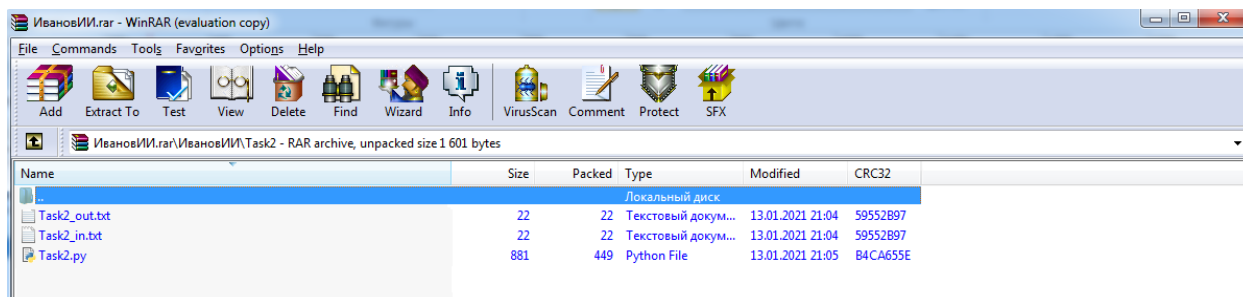


Рисунок П.3 – Пример третьего уровня содержимого архива

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.

**Задания на очный этап олимпиады «Информатика»
Вариант №2**

Задание 1 (6 баллов)

Требуется реализовать программное средство для корректного расчета функции, представленной на рисунке 1.1.

$$y = \frac{1}{\cos(x)} + \sqrt{\frac{\sin^2(x)}{x^2 - 16}}$$

Рисунок 1.1

При программной реализации следует предусмотреть отсутствие ошибок при вычислении.

Задание 2 (14 баллов)

Выполнить программную реализацию расчета значений логической функции четырех переменных (см. рис. 2.1) и вывод результата в файл Task2_out.txt.

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

Рисунок 2.1 – Логическая функция четырех переменных

В функции обозначения соответствуют:

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

Исходные данные с 16 комбинациями значений переменных требуется считать из файла Task2_in.txt. (пример см. рис. 2.2, порядок столбцов А В С D всегда соответствует данному примеру и не может быть изменен).

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

Рисунок 2.2 – Файл Task2_in.txt

Значение F вычисляется разработанной программой (за ручное решение с записью в файл Task2_out.txt баллы не начисляются), и записывается в файл Task2_out.txt.

На рисунках 2.3 а) и 2.3 б) показаны примеры файлов Task2_in.txt и Task2_out.txt, заполненных в соответствии с требованиями, для логической функции от двух переменных $F = A \text{ and } B$.

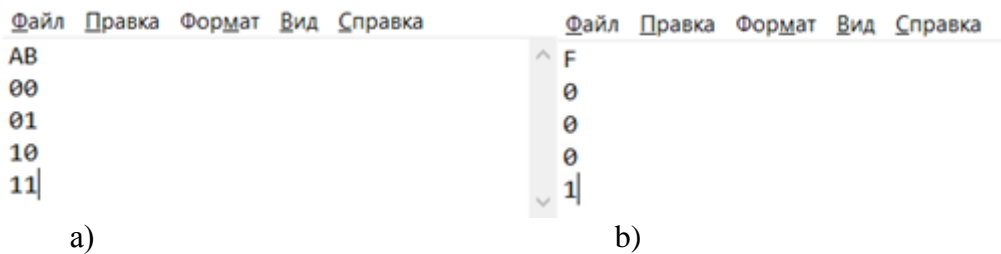


Рисунок 2.3 – а) Файл Task2_in.txt, б) Файл Task2_out.txt

Задание 3 (20 баллов)

Вячеслав и Анатолий решили использовать в своей переписке методы простого шифрования. Проанализировав длину сообщений, они выяснили, что она всегда равна 16 символам без учета пробелов, так как ребята их не используют. На основе данного анализа было принято использовать в качестве инструмента шифрования магический квадрат размерностью 4X4.

Магический или волшебный квадрат представляет из себя квадратную матрицу, заполненную различными числами таким образом, что сумма чисел в каждой строке, каждом столбце и на обеих диагоналях одинакова.

Пример магического квадрата 4X4 представлен на рисунке 3.1.

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

Рисунок 3.1

Алгоритм шифрования с использованием магического квадрата, следующий:

- ставим первую букву в ту клетку магического квадрата, где написано самое маленькое, число;
- остальные буквы (вторую и далее) на позиции с соответствующими числами (по возрастанию) (см. рис. 3.2)

Например, фраза «ПриезжаюСегодня.»

16 «.»	3 «и»	2 «р»	13 «д»
5 «з»	10 «е»	11 «г»	8 «ю»
9 «С»	6 «ж»	7 «а»	12 «о»
4 «е»	15 «я»	14 «н»	1 «П»

Рисунок 3.2

Считываем квадрат построчно начиная с верхнего левого угла и получаем зашифрованную фразу: «.ирдзегюСжаоеянП»

Магический квадрат является ключом для шифрования и расшифровки.

Помогите ребятам автоматизировать процесс шифрования и расшифровки с помощью описанного метода.

Требования к программе:

1. Пользователь выбирает действие, которое собирается осуществить (см. рис. 3.3):

Выберите действие:
 Шифрование нажмите - 1
 Дешифрование нажмите - 2

Рисунок 3.3

2. При нажатии «1» программное средство должно выполнить шифрование.
3. При нажатии «2» программное средство должно выполнить дешифрование.

При шифровании в качестве входного файла программа должна использовать файл Task3_in_SH.txt структура которого представлена на рисунке 3.4.

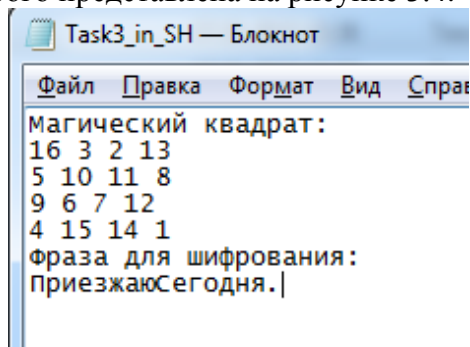


Рисунок 3.4

В качестве выходного файла формируется файл Task3_out_SH.txt структура которого представлена на рисунке 3.5.

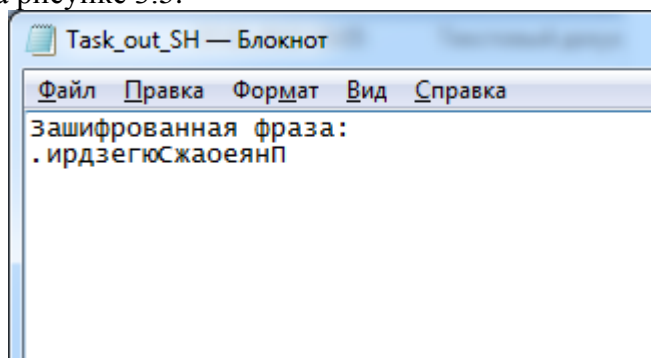


Рисунок 3.5

При дешифровании в качестве входного файла программа должна использовать файл Task3_in_DSH.txt структура которого представлена на рисунке 3.6.

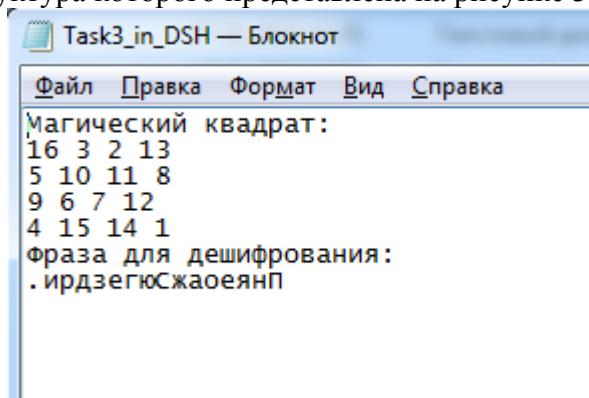


Рисунок 3.6

В качестве выходного файла при дешифровании формируется файл Task3_out_DSH.txt структура которого представлена на рисунке 3.7.

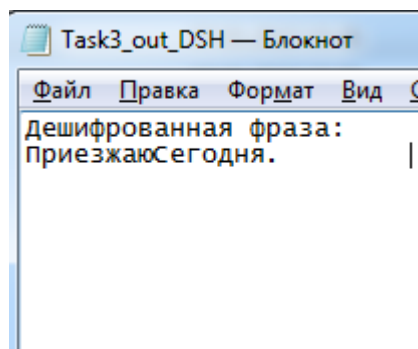


Рисунок 3.7

При шифровании и дешифровании фраз требуется выполнить проверки:

1. Длина сообщения должна соответствовать 16 символам без пробелов (пробелы в сообщениях не используются);
2. Корректность магического квадрата 4X4.

При несоответствии требованиям программное средство должно выводить пользователю соответствующее сообщение или сообщения:

1. «Длина сообщения не соответствует требованиям»;
2. «Магический квадрат 4X4 некорректен».

Задание 4 (25 баллов)

Космонавт веден наблюдение за водным пространством планеты в определенном секторе побережья с помощью специальной камеры, которая выполняет ежедневную фотосъемку. На основании данных снимков космонавт выполняет подсчет количества островов и рифов. Выполняя данную работу космонавт заметил, что если фотографии отобразить с помощью бинарной квадратной матрицы размерностью 10X10, то рифы можно обозначить одной 1, которая окружена 0 или краем/краями матрицы, а острова представляются от двух до 5 единиц, расположенных в соседних клетках, окруженных 0 или краем/краями матрицы, более 5 единиц, расположенных в соседних клетках, окруженных 0 или краем/краями матрицы, это прочие объекты, которые не требуется учитывать. Помогите космонавту автоматизировать процесс подсчета, путем разработки специальной программы, которая на основании заданного входного файла подсчитает количество рифов и островов.

В качестве входного файла программа должна использовать файл Task4_in.txt структура которого представлена на рисунке 4.1.

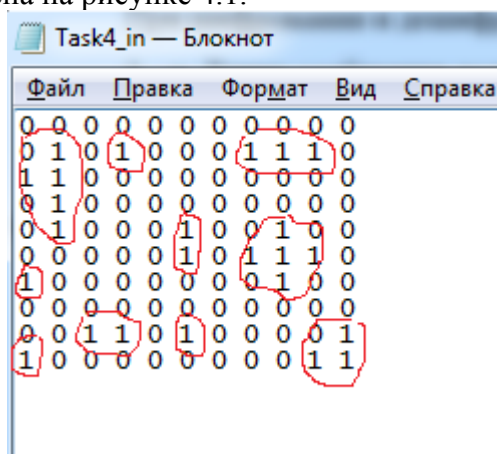


Рисунок 4.1

Как видно из примера на снимок отобразил:

Рифов – 4 шт.;

Островов – 6 шт.

Полученные данные требуется сохранить в файл Task4_out.txt, структура которого показана на рисунке 4.2.

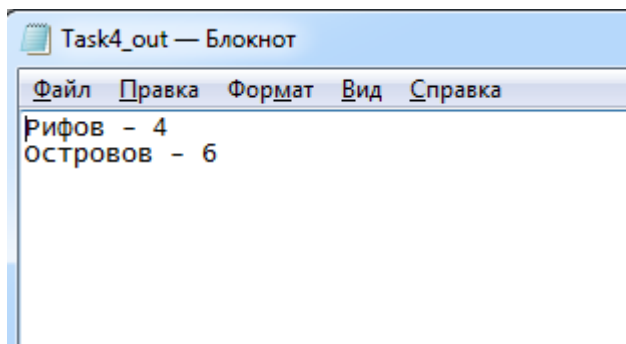


Рисунок 4.2

Задание 5 (35 баллов)

Турецкие шашки распространены в Турции, Кувейте, Ливане, Сирии, Иордании, Греции, Армении, а также в Израиле. Турецкие шашки отличаются от привычных нам как начальным положением, так и правилами ходов и взятия.

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

Правила игры:

1. Начальная позиция — 16 белых и 16 чёрных шашек занимают вторую и третью горизонтали с каждой стороны, оставляя первую свободной.

Турецкие шашки

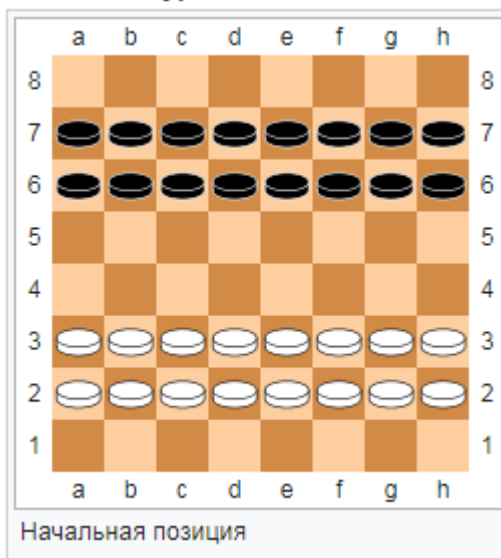


Рисунок 5.1

2. Простые шашки ходят и бьют по вертикалям и горизонталям.
3. Простая шашка ходит на одно поле вперёд, влево, вправо.
4. Простая шашка бьёт шашку противника перескакивая через неё на следующее поле по вертикали или горизонтали, назад бить запрещается.
5. Если есть несколько вариантов боя, игрок обязан выбрать тот, при котором берётся наибольшее количество шашек противника.

Для программной реализации алгоритма вычисления хода поле для игры представляется текстовым файлом. На рисунке 5.2 показано кодирование игрового поля:

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

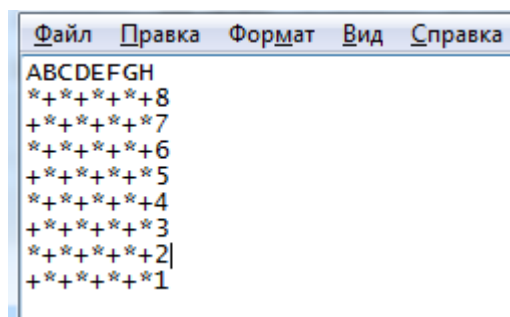


Рисунок 5.1 – Кодирование игрового поля.

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 расположения шашек и кодировки расположения шашек на доске с помощью текстового файла Task5_in.txt смотрите на рисунке 5.2

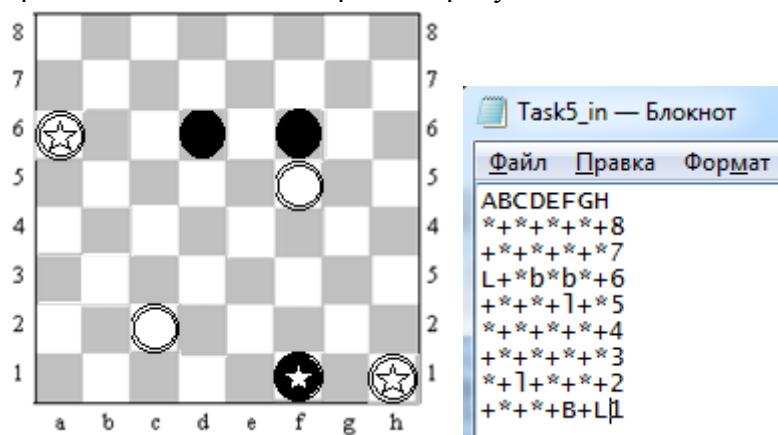


Рисунок 5.2

Запись вариантов хода осуществляется в соответствии со следующими правилами:

- для записи простого хода (без боя) обозначают сначала поле, на котором шашка (или дамка) стояла, затем ставят тире и записывают поле, на которое она ставится (пример c2-c3 или c2-d2).
- при записи взятия (боя) вместо тире ставится двоеточие (пример ab:e6);
- 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 ab:g6).
- 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 h1:d1:d7).

Задача программной реализации состоит в следующем:

- из заданного расположения шашек в файле Task5_in.txt (пример см. рис. 5.3), требуется найти ход простой шашкой в соответствии с правилами;

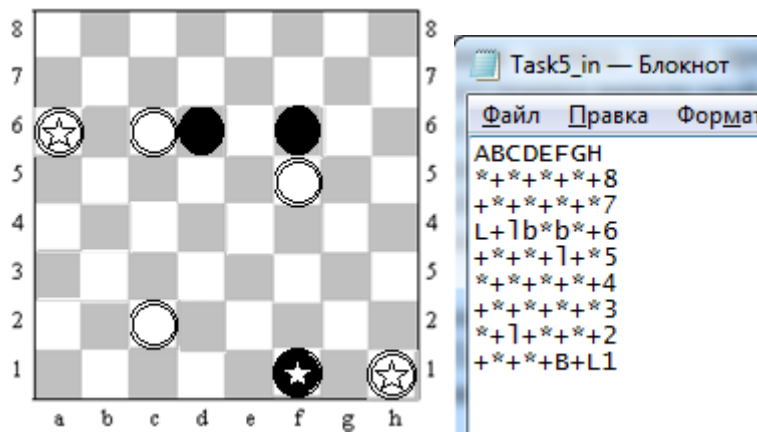


Рисунок 5.3

- право хода принадлежит белым шашкам;
- ход требуется записать в файл Task5_out.txt (см. рис. 5.4);

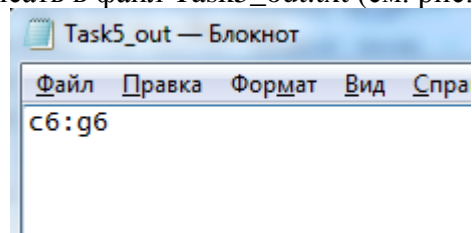


Рисунок 5.4

- если количество ходов, более одного, и они могут быть осуществлены, в соответствии с правилами турецких шашек, т. е. у игрока есть выбор, то ходы записываются в столбик;
- если взятие невозможно, требуется рассчитать простой ход;
- если ход невозможен, то требуется осуществить запись «No» в файл Task5_out.txt.