Задания на очный этап олимпиады «Информатика»
Вариант №1

Задание 1 (8 баллов)

Реализовать программу для расчета функции y=y(x), представленной на рисунке 1.1. Значения x вводятся с клавиатуры, результат вычислений выводится на экран ПК.

$$y = \frac{x}{\sqrt{x+4}} + log_2(x+1)$$

Рисунок 1.1 — Функция y=y(x)

Задание 2 (12 баллов)

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

$$F = (A\ and\ notB)\ \text{or}\ (C \rightarrow D)\ and\ B$$

Рисунок 2.1 – Логическая функция четырех переменных

В функции обозначения соответствуют:

- *or* – логическое ИЛИ (дизъюнкция);
- and – логическое И (конъюнкция);
- *not* – логическое отрицание (НЕ);
- → – импликация;

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

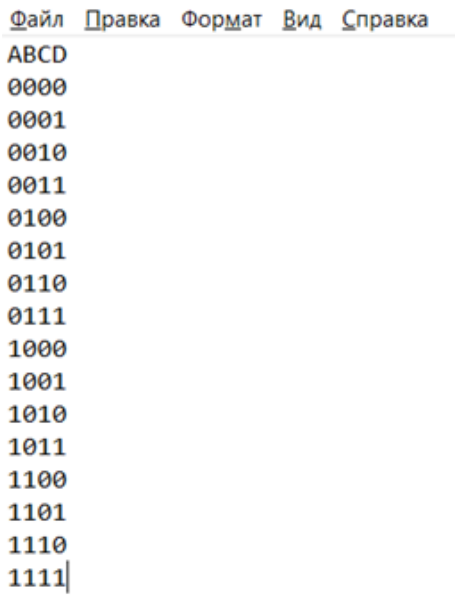
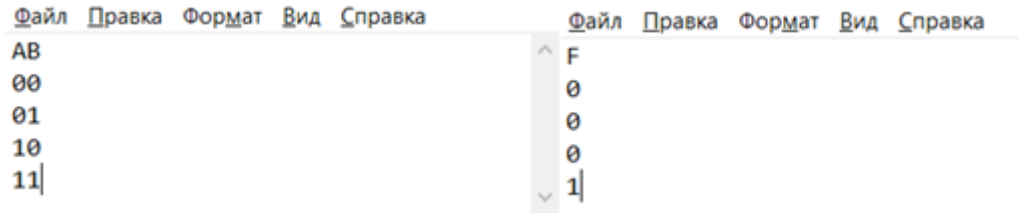


Рисунок 2.2 – Файл Task2_in.txt

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 a) и 2.3 b) показаны примеры файлов Task2_in.txt и Task2_out.txt,

заполненных в соответствии с требованиями, для логической функции от двух переменных F = A and B.



a) b)

Рисунок 2.3 – a) Файл Task2_in.txt, b) Файл Task2_out.txt

Задание 3 (20 баллов)

В переписке между учебными комплексами Финуниверситета для передачи важных сообщений решили использовать шифр. Длина сообщений при этом никогда не превышает 25 символов, включая пробелы.

Процесс шифрования начинается с построения сетки размером 5×5, где каждая ячейка заполняется 25 буквами латинского алфавита (буквы I и J шифруются как I). Каждая строка и столбец сетки задаётся одной из 5 цифр: «1», «2», «3», «4» и «5».

	1	**2**	**3**	**4**	**5**
1	A	B	C	D	E
2	F	G	H	I	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Рисунок 3.1

Рассмотрим процесс шифрования на примере небольшого сообщения: «SOMETEXT».

На первом шаге каждый символ сообщения заменяется на пару координат, обозначающих строку и столбец соответствующего символа в сетке:

Сообщение	S	O	M	E	T	E	X	T
Координата по вертикали	3	4	2	5	4	5	3	4
Координата по горизонтали	4	3	3	1	4	1	5	4

Рисунок 3.2

На втором шаге координаты считываются по строкам:
3425453443314154.

На третьем шаге полученная последовательность сдвигается циклически влево на один шаг:
4254534433141543.

На четвертом шаге координаты преобразуются в буквы по сетке, представленной на рисунке 3.1:

Координата по вертикали	4	5	5	4	3	1	1	4
Координата по горизонтали	2	4	3	4	3	4	5	3
Сообщение	I	U	P	T	N	Q	V	O

Рисунок 3.3

Окончательный вид шифротекста: IUPTNQVO.

Для восстановления исходного текста необходимо выполнить действия, обратные шифрованию.

Необходимо автоматизировать процесс шифрования и дешифрования с помощью описанного метода (пробелы в сообщениях не шифруются и соответственно не дешифруются).

Требования к программе:

1. Пользователь выбирает действие, которое собирается осуществить (реализация меню см. рис. 3.4):

```
Выберите действие:
Шифрование нажмите - 1
Дешифрование нажмите - 2
```

Рисунок 3.4

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 3.5.

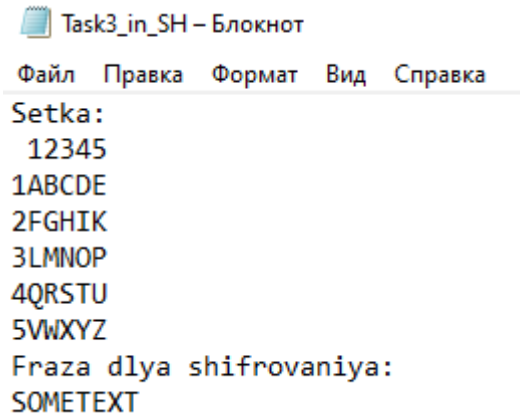


Рисунок 3.5

В качестве выходного файла формируется файл Task3_out_SH.txt структура которого представлена на рисунке 3.6.

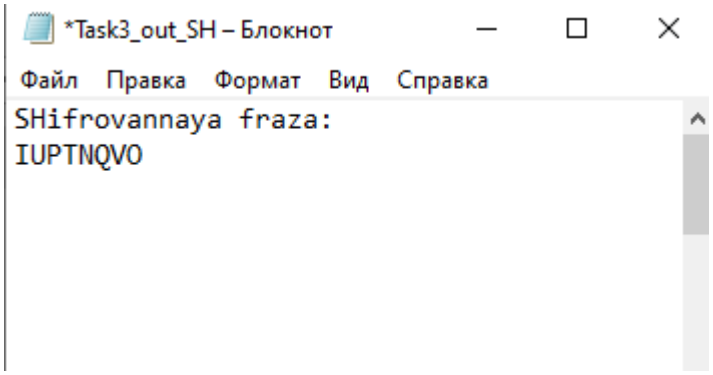


Рисунок 3.6

При дешифровании в качестве входного файла программа должна использовать файл Task3_in_DSH.txt структура которого представлена на рисунке 3.7.

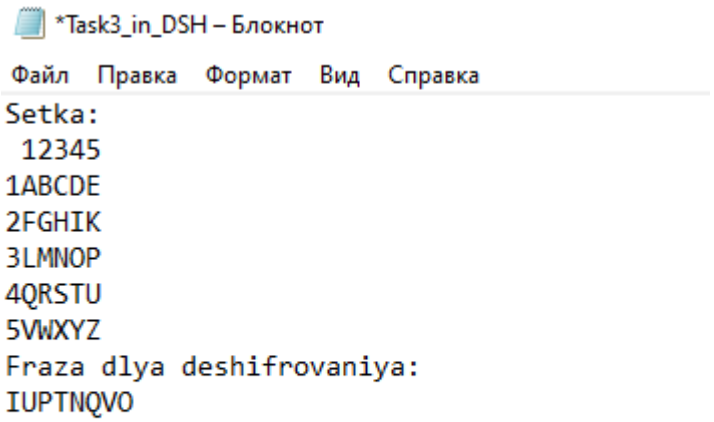


Рисунок 3.7

В качестве выходного файла при дешифровании формируется файл Task3_out_DSH.txt структура которого представлена на рисунке 3.8.

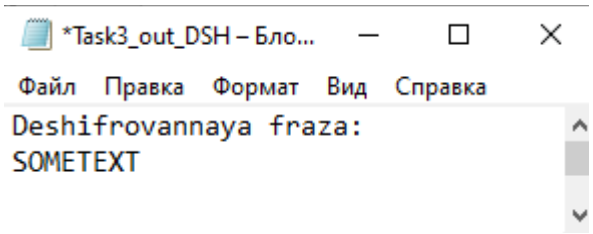


Рисунок 3.8

При шифровании и дешифровании фраз требуется выполнить проверки:

1. На существование в исходном сообщении всех символов, которые будут подвержены шифрованию или дешифрованию (сообщение должно содержать только буквы латинского алфавита и пробелы);
2. Длина сообщения не превышает 25 символов (пробелы из сообщения удаляются или сообщение вводится без пробелов).

При несоответствии требованиям программное средство должно сформировать следующие сообщениях в Task3_out_SH и Task3_out_DSH:

1. «В исходном сообщении указаны символы, которые не могут быть зашифрованы или дешифрованы»;
2. «Длина сообщения не соответствует требованиям».

Задание 4 (25 баллов)

При исследовании систем, состоящих из множества взаимосвязанных объектов, широко используются графовые модели. Одним из способов описания графа является матрица смежности – квадратная таблица, в которой значение в позиции (i, j) показывает, существует ли связь между объектами i и j. В данной задаче рассматриваются только неориентированные графы, поэтому корректная матрица смежности должна удовлетворять нескольким обязательным свойствам: она должна быть симметрична относительно главной

диагонали, содержать только нули и единицы, а элементы на диагонали должны быть равны нулю (это означает отсутствие петель – связи вершины с самой собой). Нарушение любого из этих свойств говорит о том, что данные содержат ошибку или записаны некорректно.

Необходимо создать входной файл **Task4_in.txt**, содержащий несколько квадратных матриц смежности одного размера. В первой строке файла должны быть записаны два натуральных числа K и N: количество матриц и их размерность соответственно. Далее в файле помещаются K последовательных блоков, каждый из которых состоит из N строк и содержит ровно N целых чисел в каждой строке. Таким образом, файл описывает K матриц размера $N \times N$, записанных друг за другом без дополнительных разделителей. Все данные формируются вручную, и при проверке программа должна строго следить за тем, чтобы количество строк и столбцов совпадало с заявленным, а все значения являлись целочисленными.

После чтения данных необходимо сначала убедиться, что формат файла корректен: первая строка действительно содержит два натуральных числа; далее присутствует ровно K блоков по N строк; в каждой строке находится ровно N значений; каждое значение может быть преобразовано в целое число. Если любое из этих условий нарушается, программа должна вывести сообщение «*Ошибка формата данных*» и завершить работу без создания выходных файлов. В случае корректности входного формата следует последовательно проверить каждую матрицу на соответствие описанным требованиям. Если матрица содержит дробные значения, отрицательные числа или любые другие значения, отличные от 0 и 1; если она несимметрична; если на диагонали встречаются ненулевые элементы – такая матрица считается *ошибочной*. Матрицы, полностью удовлетворяющие требованиям, относятся к *корректным*.

Результаты проверки необходимо разделить на две группы. Все корректные матрицы должны быть записаны в файл **Task4_1_out.txt**. В его первой строке указывается количество корректных матриц K_1 и размерность N, а далее, без пропусков и дополнительных комментариев, последовательно приводятся все K_1 матриц в том же формате, в котором они были даны во входном файле. Если корректных матриц нет, файл должен содержать только строку «*0 N*». Аналогично, файл **Task4_2_out.txt** должен включать все ошибочные матрицы. Его первая строка имеет вид «K_2 N», где K_2 – количество ошибочных матриц. Сразу после каждой ошибочной матрицы должна приводиться строка, в которой в свободной форме перечисляются выявленные нарушения, например: «*Ошибки: нарушение симметрии; неверная диагональ*» или «*Ошибки: недопустимые значения*». Если ошибочных матриц нет, файл также должен состоять из единственной строки «*0 N*». По завершении корректной работы программа должна вывести на экран содержимое исходного файла и обоих выходных файлов.

На рис. 4.1 приведён пример корректного входного файла и соответствующих выходных данных.

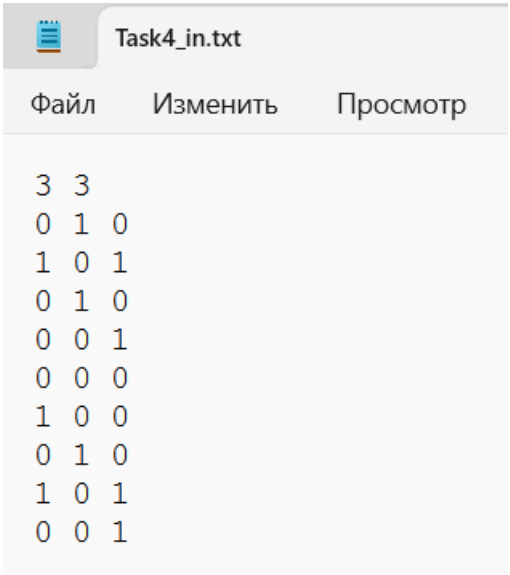


Рисунок 4.1 – Пример входного файла Task4_in.txt

Первая из представленных матриц является корректной: она содержит только допустимые значения 0 и 1, её диагональные элементы равны нулю, а сама матрица симметрична относительно главной диагонали. Вторая матрица также корректна и полностью соответствует требованиям, предъявляемым к матрицам смежности неориентированного графа. Третья матрица является ошибочной: в ней нарушено условие симметричности и на диагонали присутствует недопустимое значение. Таким образом, две первые матрицы считаются корректными, а третья – ошибочной.

В результате выполнения программа должна сформировать следующие файлы (рис. 4.2 и 4.3).

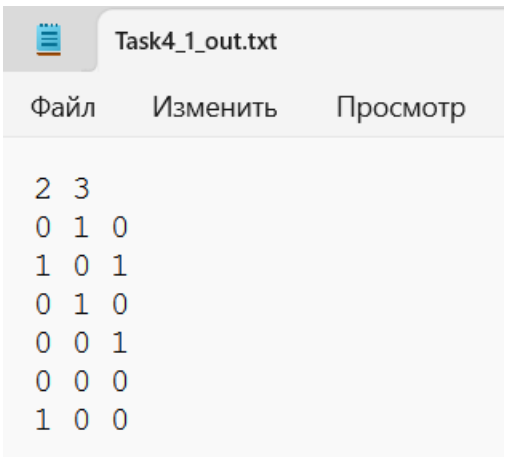


Рисунок 4.2 – Файл Task4_1_out.txt

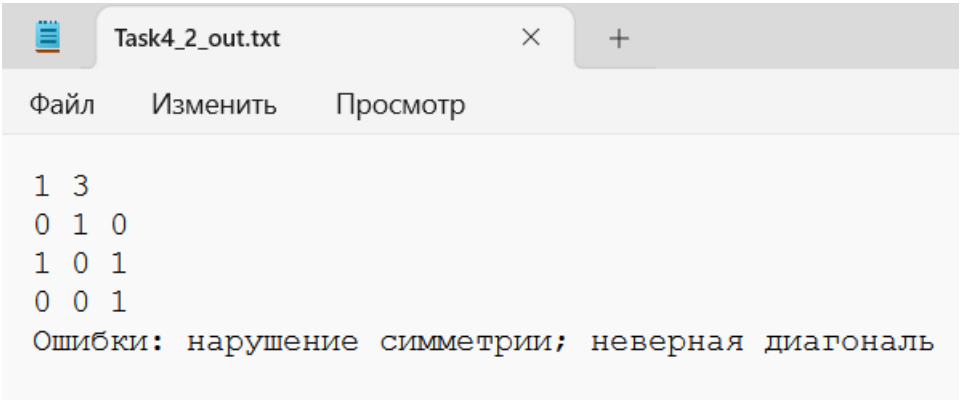


Рисунок 4.3 – Файл Task4_2_out.txt

Теперь приведём пример (рис. 4.4), в котором ошибка возникает уже на этапе чтения формата. Если, например, во входном файле отсутствует одна из строк матрицы или строка содержит неправильное количество чисел, программа должна немедленно завершить работу, выведя сообщение «*Ошибка формата данных*», и не создавать выходные файлы.

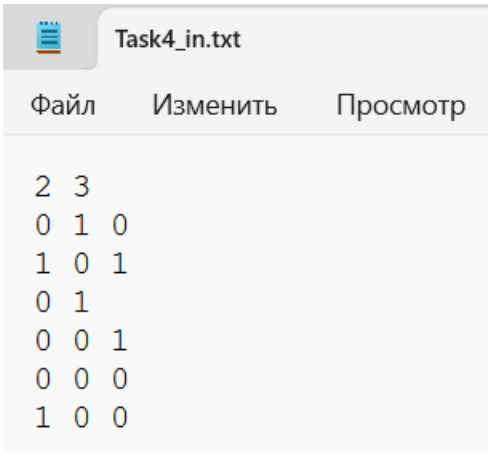


Рисунок 4.4 – Некорректный входной файл

В приведённом примере нарушена структура входных данных: одна из строк матрицы содержит два числа вместо требуемых трёх. В таких случаях программа должна завершить работу и вывести сообщение «*Ошибка формата данных*», не создавая никаких выходных файлов.

Следующий пример иллюстрирует ситуацию, когда ни одна из матриц не удовлетворяет требованиям корректности (рис. 4.5):

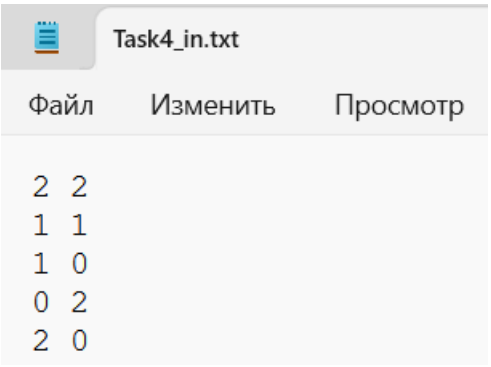


Рисунок 4.5 – Пример входного файла с ошибочными матрицами

Первая матрица некорректна из-за ненулевых диагональных элементов, вторая – из-за наличия элементов вне допустимого диапазона. В результате файл Task4_1_out.txt будет состоять из единственной строки (рис. 4.6).

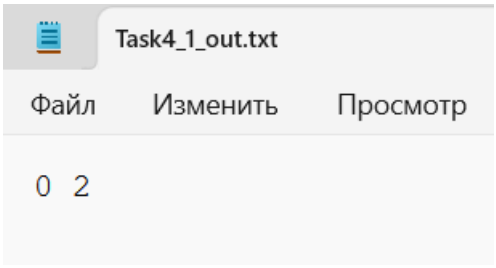


Рисунок 4.6 – Пример файла Task4_1_out.txt при отсутствии корректных матриц

Файл Task4_2_out.txt в этом случае должен содержать обе ошибочные матрицы, каждая из которых сопровождается описанием выявленных нарушений (рис. 4.7).

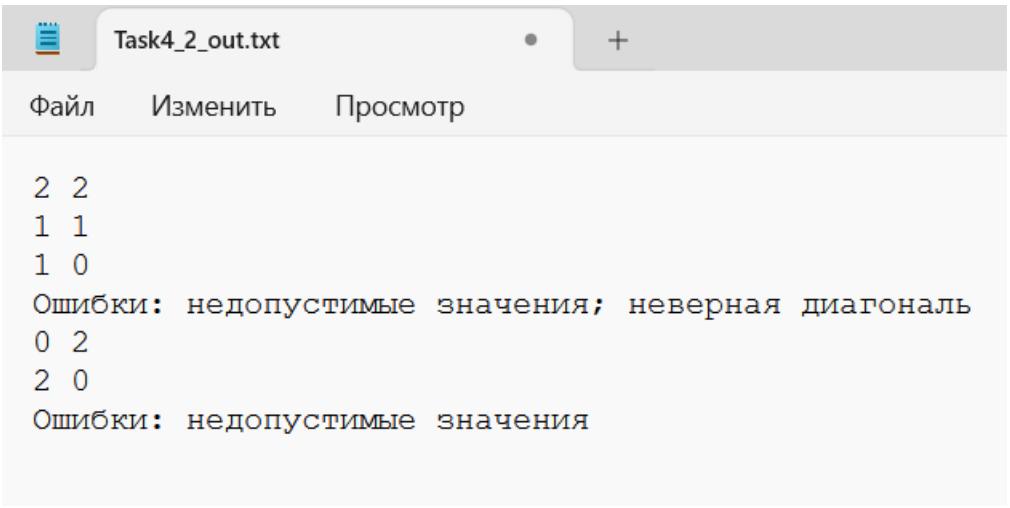


Рисунок 4.7 – Пример файла Task4_2_out.txt с ошибочными матрицами

Задание 5 (35 баллов)

Напишите программу, которая определяет фигуру, создавшую положение в шахматной партии «двойной удар» после хода белых.

Условия шахматной ситуации:

очередной ход сделан белыми;

на шахматной доске может находится любое допустимое количество черных и белых фигур (пешка, слон, конь, ладья или ферзь), черный король находится обязательно.

Определение:

«Двойной удар» в шахматах или, «вилка» — положение в шахматной партии, когда две или более фигуры одного игрока находятся под боем одной фигуры другого игрока. (пример «двойного удара» на рисунке 5.1).

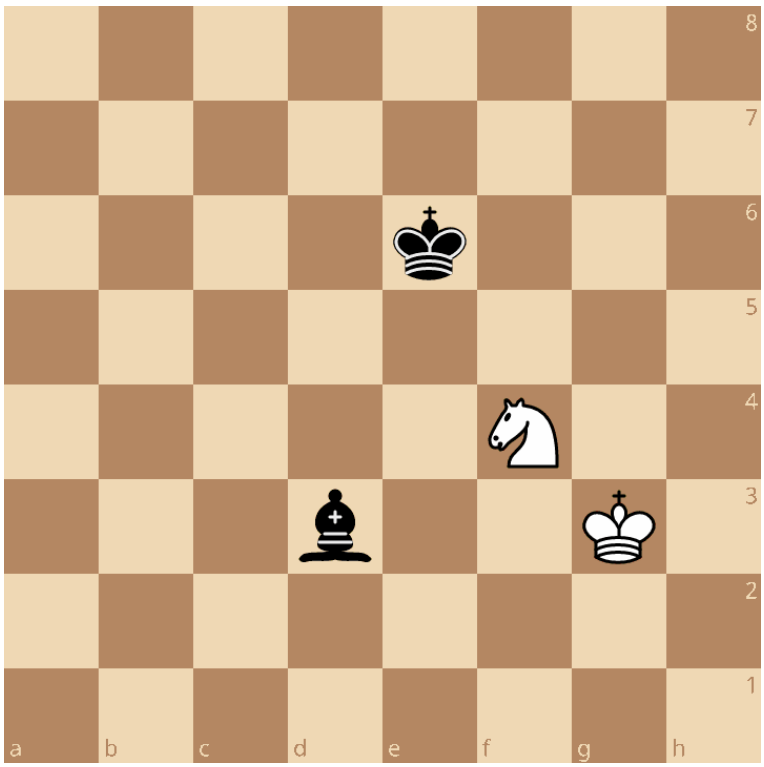


Рисунок 5.1 – Пример положения в шахматной партии «двойной удар»

Обозначения:

Игровое поле для игры в шахматы представляет собой доску, разделенную на 64 квадратные клетки по 8 клеток по горизонтали и 8 клеток по вертикали. Каждая вертикаль обозначается латинской буквой от a до h, а каждая горизонталь - арабской цифрой от 1 до 8. Таким образом каждая клетка шахматной доски обозначается двумя символами – буквой вертикали и цифрой горизонтали, к которым принадлежит эта клетка, например b2 или e4 (представлена на рисунке 5.2).

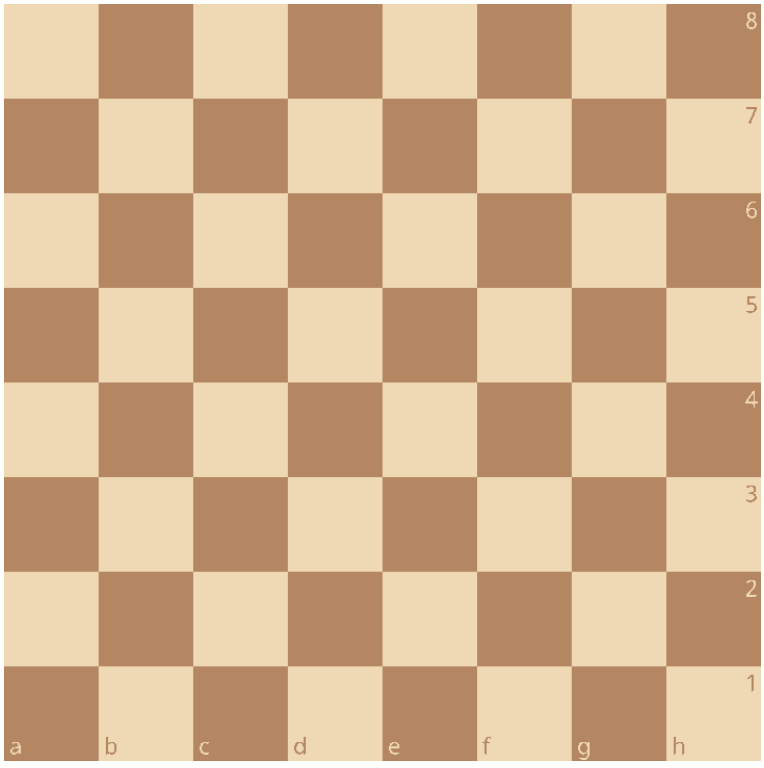


Рисунок 5.2 – Шахматная доска

Шахматная фигура может находится на одной определенной клетке доски. На одной клетке может находится только одна фигура. Для записи положения фигуры будет использоваться стандартное обозначение - имя фигуры и имя клетки, на которой располагается фигура. Фигуры обозначаются так: N - конь, B - слон, R - ладья, Q - ферзь, K - король. Пешка никак не обозначается. Так, например, запись Nb1 означает, что конь находится на клетке b1, запись e8 означает, что пешка находится на клетке e8. Пробел между именем фигуры и именем клетки не ставится.

Правила игры:

Пешки могут двигаться только на одну клетку по вертикали. Белые пешки двигаются «вверх» - в направлении увеличения числа в имени клетки, а черные – «вниз» - в направлении уменьшения. Назад пешки ходить не могут. Исключение - белые пешки, находящиеся на 2 горизонтали, и черные - на 7 горизонтали. Эти пешки могут ходить как на одну клетку вперед, так и на две.

Ладья движется по горизонтали и вертикали на любое доступное количество клеток.

Слон движется по горизонтали на любое доступное количество клеток.

Ферзь может перемещаться на любое количество клеток по горизонтали, вертикали или диагонали.

Конь ходит буквой «Г»: две клетки в одном направлении (горизонтально или вертикально) и затем одну клетку в сторону.

На рисунке 5.3 кружком отмечены клетки, на которые могут переместиться соответствующие фигуры.

Обратите внимание, что фигуры не могут выходить за пределы доски.

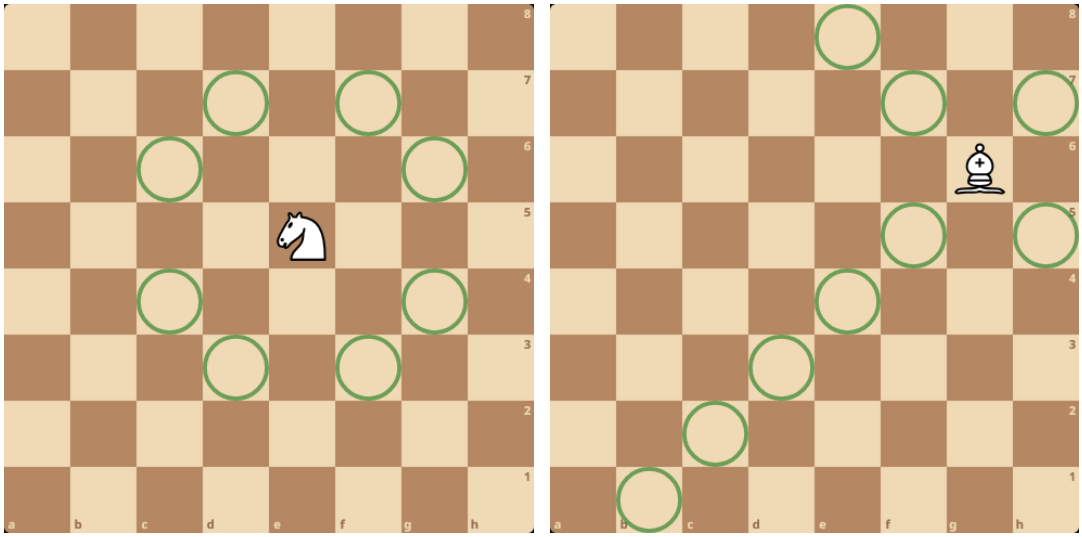


Рисунок 5.3 – Возможные варианты хода фигур «Конь» и «Слон»

Король перемещается на одну клетку в любом направлении (горизонтально, вертикально или по диагонали).

Фигуры не могут двигаться за пределы доски. Фигуры за исключением коня не могут «перепрыгивать» через другие фигуры, в том числе короля.

Шах - это ситуация, когда король находится на клетке под боем фигуры или пешки противоположного цвета. Клетка считается под боем фигуры тогда, когда эта фигура может следующим ходом переместиться со своего текущего положения на эту клетку. Исключение - пешки. Под боем пешки находятся клетки по диагонали.

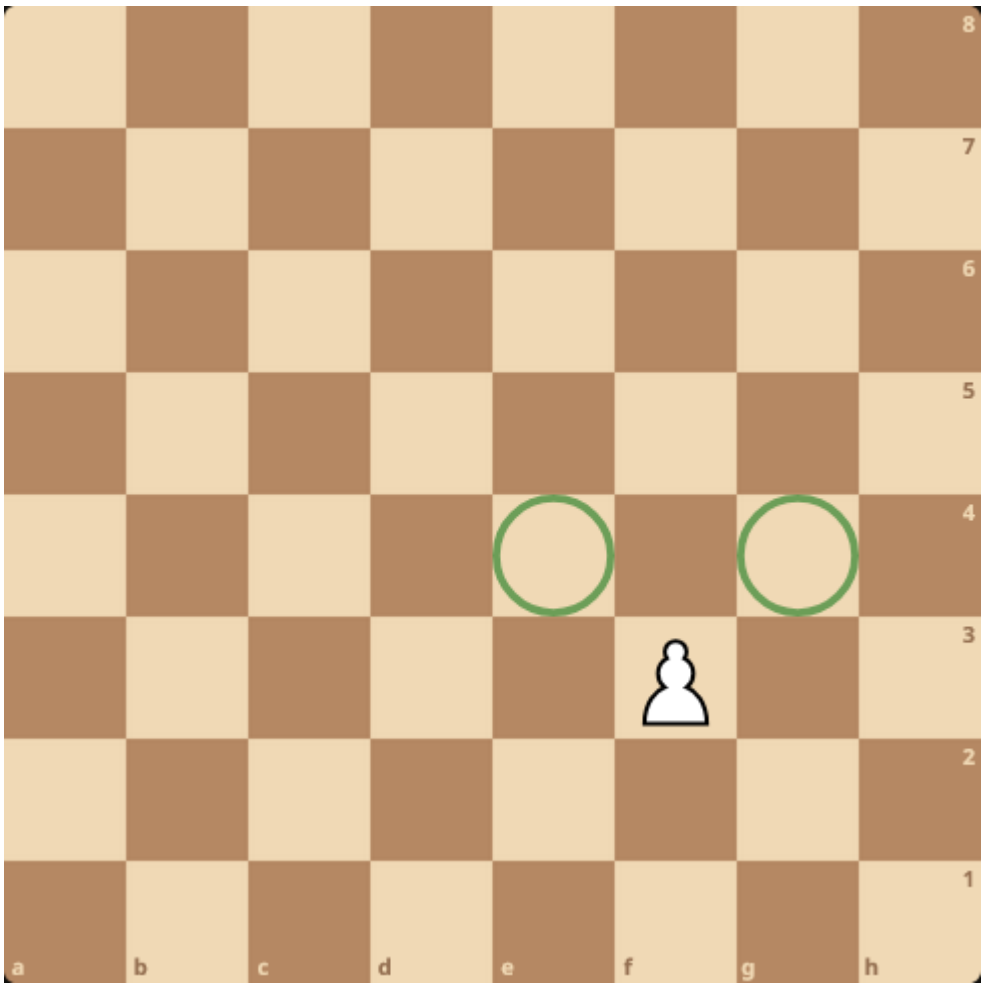


Рисунок 5.4 – Клетки под боем фигуры пешка

На рисунке 5.4 кружком обозначены клетки, находящиеся под боем пешки. Обратите внимание, что направление боя для пешки зависит от ее цвета и, как следствие, направления движения.

Формат входного файла:

Входной файл состоит из двух строк:

- в первой строке обязательно указывается положения черного короля и как минимум одной из черных фигур, разделенных пробелом;
- во второй строчке указываются положения белых фигур, количество фигур равно одному и более, разделённые пробелом.

Пример файла:

c7 Kc8 Re4

Kf1 Nc5

соответствует такому положению на доске (рисунок 5.5):

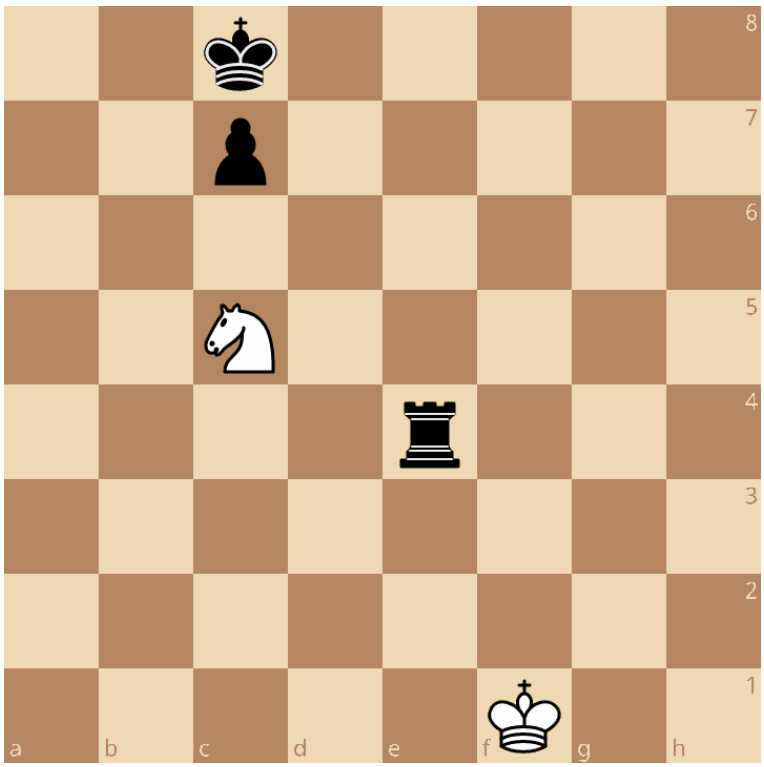


Рисунок 5.5 – Положение фигур

А такой файл:
Rc8 Ke8 Nh7
Ke1 Qf5
соответствует такому положению на доске (рисунок 5.6):

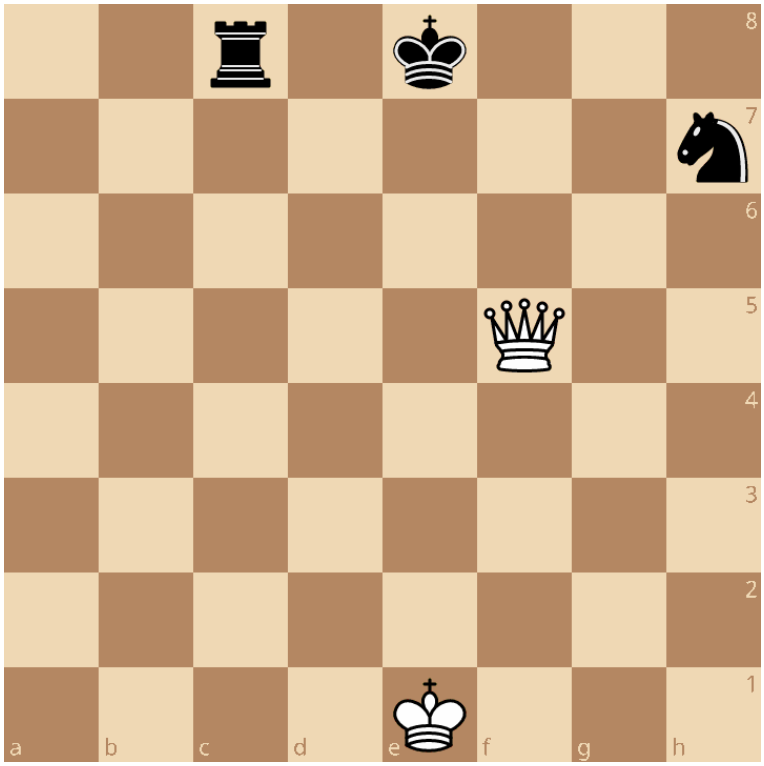


Рисунок 5.6 – Положение фигур

Формат выходного файла:

Если ситуация на шахматной доске определена как «двойной удар», то программа должна записать в выходной файл белую фигуру, создавшую эту ситуацию, и ее позицию на доске, иначе в выходном файле необходимо разместить значение «0».

Первый пример:
Входной файл:
c7 Kc8 Re4
Kf1 Nc5
Выходной файл должен содержать: 0.

Второй пример:
Входной файл:
Rc8 Ke8 Nh7
Ke1 Qf5
Выходной файл должен содержать: Qf5.