**Лабораторная работа №1**

**Цель:** При работе над данной лабораторной работой затронуть следующие вопросы:

1. Работа с текстовыми файлами,
2. Работа с локальными и глобальными переменными,
3. Работа с несколькими функциями, а не только с main,
4. Нисходящий метод разработки программ,
5. Анализ ходов.

**Ход работы:**

**Задача**: Дана квадратная шахматная доска размером N x N. На доске уже размещено K фигур. Фигуры размещены так, что находятся не под боем друг друга.

Необходимо расставить на доске еще L фигур так, чтобы никакая из фигур на доске не находилась под боем любой другой фигуры. **Необходимо найти все возможные решения.**

Вы работаете с фигурой: *index 4 (BishopTwoCellsFigure)*

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 \* 0 0 0 \* 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 \* 0 \* 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 # 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 \* 0 \* 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 \* 0 0 0 \* 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Входные данные в файле input.txt. На первой строке файла записаны три числа: N L K (через пробел). Далее следует K строк, содержащих числа x и y (через пробел) - координаты уже стоящей на доске фигуры (фигуры стоят правильно). Координаты отсчитываются от 0 до N-1. 1 <= N <= 20.

**Ваш алгоритм должен быть оптимизирован, для работы с доской 20!!**

Выходные данные в файл output.txt. **На каждое найденное решение необходимо записать в файл одну строку**. Строка состоит из пар (x,y) - координаты фигур на доске. **В решение следует вывести координаты всех фигур, находящихся на доске.** Каждую фигуру необходимо записать в виде пары координат, разделенных запятой и обрамленных скобками. Координаты отсчитываются от 0 до N-1. Порядок, в котором фигуры перечислены в решении, не имеет значения. Если не было найдено ни одного решения, в файл необходимо записать «*no solutions*».

Выходные данные на консоль — это доска N\*N, где фигура обозначается #, ее ходы обозначаются \*, а пустые клетки обозначаются 0.

*Фигура BishopTwoCellsFigure. input:*

20 0 1

8 8

**Решение**:

def print\_board(N, board): # Функция для отображения доски. Чтобы Доска отобразилась в консоли, впишите до "\_\_main\_\_" - "print\_board(N,board)"  
 for i in range(N):  
 row=''  
 for j in range(N):  
 if board[i][j]==1: row+='# '  
 elif board[i][j]==2: row+='\* '  
 else: row+='0 '  
 print(row.strip())  
  
def mark\_attacks(N, board, x, y): # Функция для пометки клеток, которые атакует фигура. Атакует только две ближайшие диагональные клетки.  
 directions = [(-1,-1),(-1,1),(1,-1),(1,1)] # Направления для фигуры (диагонали)  
 for dx, dy in directions: # Для каждой диагонали проверяем две ближайшие клетки  
 # Первая клетка  
 nx, ny = x + dx, y + dy  
 if 0<=nx<N and 0<=ny<N: board[nx][ny]=2 # Помечаем как атакуемую клетку  
 # Вторая клетка  
 nx, ny = x + 2 \* dx, y + 2 \* dy  
 if 0<=nx<N and 0<=ny<N: board[nx][ny]=2 # Помечаем как атакуемую клетку  
  
def main():  
 with open('input.txt', 'r') as f: # Чтение входных данных из файла  
 N,L,K = map(int,f.readline().split())  
 already\_placed = [tuple(map(int, f.readline().split())) for \_ in range(K)] # Позиции уже стоящих фигур  
  
 board = [[0] \* N for \_ in range(N)] # Инициализация доски  
 for x,y in already\_placed: board[x][y]=1 # Ставим фигуру на доску  
 for x,y in already\_placed: mark\_attacks(N,board,x,y) # Для каждой фигуры на доске помечаем её ходы  
  
 with open('output.txt', 'w') as f: # Запись в файл в формате "x y"  
 if already\_placed:  
 solution\_str = '\n'.join(f'{x} {y}' for x,y in already\_placed)  
 f.write(solution\_str+'\n')  
 else: f.write("no solutions\n")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**Пример:**

|  |  |
| --- | --- |
| Input: 20 0 1 8 8 | Output: 8 8 |

**Вывод:** В ходе лабораторной работы затронули следующие работы:

1. Работа с текстовыми файлами,
2. Работа с локальными и глобальными переменными,
3. Работа с несколькими функциями, а не только с main,
4. Нисходящий метод разработки программ,
5. Анализ ходов.