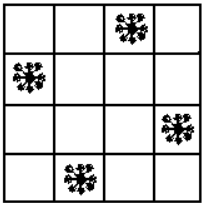
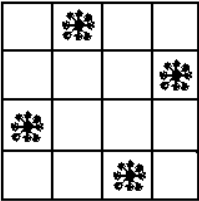
**Лабораторное задание №3**

**Описание задачи:**

На шахматной доске с заданным размером **n×n** клеток, необходимо расставить **n** ферзей так, чтобы каждый ферзь не атаковал друг друга. Сколько комбинаций такой расстановки возможно для размера доски от **4×4 и** до **15×15**?

Для доски размером **4×4** таких решений **2**:

Программа ***nqueens.cpp*** последовательно решает данную задачу для размеров доски от **1×1 и** до **15×15** за счет рекурсивного запуска функции ***setQueen***.

**Задание:**

1. Необходимо собрать проект с исходным файлом ***nqueens.cpp*** и запустить собранный исполняемый файл. Оценить время работы программы и корректность ее работы. Зафиксируйте это в отчете.

2. С помощью инструментария Advisor необходимо получить метрики исполнения программы в последовательном режиме: GFLOPs, количество используемых потоков, тип векторных команд, построить roofline, сохранить snapshot.

3. Максимально ускорить выполнение программы, убедиться в корректности ее работы, построить совместные roofline для разных решений, оценить во сколько произошло ускорение. Для успешной сдачи ЛР необходимо добиться хотя бы 2-х кратного ускорения выполнения программы.

4. Подготовьте отчет (разделы ниже) и продемонстрируйте работу приложений и их характеристики преподавателю.

**Подготовьте отчет со следующими разделами:**

1. Опишите каким образом Вы проводили оптимизацию последовательно исполняемого приложения (смена последовательности выполнения циклов, разбивка циклов на подциклы, использование SoA вместо AoS и наоборот, раскручивание циклов). Оцените влияние последовательности доступа к данным и их выравнивания в памяти на векторизацию.
2. Какие значения основных метрик производительности последовательно исполняемого приложения в Intel Advisor Вы получили (производительность в GFLOPs, пропускная способность памяти, использование векторных инструкций, roofline модель). Как изменились метрики при введении векторизации? Представьте сравнение метрик производительности последовательной работы с приложением, в котором используется «ручная» оптимизация.
3. Проведите прогнозирование ускорения выполнения программы за счет введения annotations в Intel Advisor. Осуществите введение параллелизма в последовательно исполняемую программу с использованием программной модели OpenMP, проведите анализ утилизации ресурсов процессора в Intel VTune, сравните нескольких roofline между собой (параллельное исполнение (OMP против ручного распараллеливания), последовательное исполнение, «ручная» оптимизация) и подтверждение выполнения прогноза ускорения в Intel Advisor. Проверьте корректность выполнения параллельной программы.