Ответ на 1 вопрос (Task1.cpp):

bool isEven(int value) {return value&1;}

Плюсы первой реализации будут в простоте и удобстве, однако она проигрывает в скорости и необходимых ресурсах процессора по сравнению со второй реализацией т.к. выполняет деление.

А вторая реализация работает быстрее 1й реализации и требует меньше ресурсов процессора, но менее удобна, т.к. работа происходит с числами в двоичной системе счисления.

Ответ на 2 вопрос (Task2.cpp):

Я решил сделать 2 реализации через массив и через vector. Можно использовать list вместо vector, но это при условии, что: важна простота добавления и удаления элементов без необходимости распределении памяти, а произвольный доступ не важен.

Плюсы реализации через массив: 1) простота 2) время доступа О(1) для методов push и pop 3) хорошая производительность  
Минусы реализации через массив: 1) фиксированный размер (если нет чёткого обозначения количества элементов) 2) утечка памяти при неправильном использовании (не освободить память)

Плюсы реализации через vector: 1) автоматическое управление памятью и динамическое изменение размера  
Минусы реализации через vector: 1) может быть менее эффективен

Ответ на 3 вопрос (Task3.cpp):

Одной из быстрых сортировок и весьма практичных является поразрядная сортировка (lsd-версия). Она стабильна, проста в реализации и очень быстра, однако не основана на сравнениях. А также из минусов можно отметить: 1) данная сортировка эффективна только для целых чисел 2) сортировка эффективна, когда количество разрядов не очень велико. Время работы составляет O(nk).

Если брать из основанных на сравнениях, то лучше всего подходит быстрая сортировка (сортировка Хоара). Эта сортировка является самой практичной сортировкой, т.к. она имеет множество оптимизаций, гибридный алгоритм и поддержку многопоточности. Однако минусами данной сортировки являются: 1) Время в лучшем и среднем варианте O(N\*logN), а в худшем варианте O(N^2) 2) Могут быть проблемы с памятью (она использует рекурсию и это может привести к переполнению стека).

Как итог хочу сказать, что есть несколько очень быстрых сортировок, однако они очень сильно зависят от условий для сортировки: тип данных, размер массива, состояние массива.