1) В каких ситуациях применяются типы std::pair и std::tuple?

std::pair и std::tuple используются для работы (хранения, обработки, вывода) с двумя/произвольным по объему множеством элементов различных типов.

2) Когда следует использовать контейнер std::array?

std::array (обертку обычного массива) следует использовать в том случае, когда ведется работа с заранее известным и не слишком большим числом переменных одного типа. Он быстродейственнее, чем std::vector, а также позволяет работать с собой в compile time.

3) Когда следует использовать контейнер std::vector?

std::vector (обертку динамического массива) следует использовать по умолчанию при работе с однотипными переменными. Позволяет эффективно работать со своим концом (увеличивать/сокращать размер, добавлять/удалять крайние элементы).

4) Когда следует использовать контейнер std::deque?

std::deque (двухстроннюю очередь) следует использовать в тех случаях, когда нужно добалять/удалять однотипные элементы не только в конце, но и в начале.

5) Когда следует использовать контейнер std::list?

std::list (двусвязный список) следует использовать тогда, когда при работе с однотипными данными необходимо добавлять/удалять элементы в произвольных местах массива, а поиск элементов для их чтения не является часто проводимой операцией.

6) Когда следует использовать контейнер std::forward_list?

std::forward_list (односвязный список) отличается от std::list лишь тем, что его элементы не содержат указателей на предыдущий элемент и предпочтительнее тогда, когда необходимость сэкономить память стоит гораздо острее, чем необходимость в процессе работы обходить элементы в обратную сторону.

7) Какие адаптеры контейнеров есть в стандартной библиотеке?

1) std::stack (первым удаляется последний добавившийся элемент).
2) std::queue (первым удаляется элемент, добавившийся раньше других).
3) std::priority_queue (очередь, в которой попутно производится сортировка).

8) Когда следует использовать контейнер circular_buffer из Boost?

boost::circular_buffer следует использовать тогда, когда для анализа потока данных требуется помнить определенное количество переменных, поступивших последними, а устаревшие данные при этом не представляют интереса. За счет перезаписывания в ячейки новых значений вместо старых происходит экономия памяти.

9) Почему контейнер circular_buffer не может войти в стандарт?

boost::circular_buffer огранизован так, что последний элемент может иметь меньший адрес, чем первый, чего стандарт для соответствующих итераторов не допускает.

10) Какие типы данных для работы с многомерными массивами вы можете назвать?

1) T[N][M] – обычный + динамический (через new-new) массивы.
2) std::vector <std::vector <T>> либо другие обертки, например, std::array<std::array<T, M>, N>.
3) boost::multi_array <T, 2U> array (boost::extents[N][M]) – возможность, предоставляемая Boost.
Примеры соответствуют двумерному случаю, в многомерном все аналогично.