1. (10 б.) Расскажите о следующих итераторах: std::back\_inserter, std::istream\_iterator, std::ostream\_iterator. Для чего они нужны, в каких случаях применяются?

back\_inserter - шаблон функции, который создает итератор для контейнера c типом, выведенным из типа аргумента – применяется, в частности, для вставки в конец контейнера.

std::istream\_iterator - это однопроходный итератор ввода, который считывает последовательные объекты – применяются для быстрого ввода с консоли

std::ostream\_iterator - это однопроходный итератор вывода, который записывает последовательные объекты – применяются для быстрого вывода в консоль

1. (10 б.) Что лежит в основе std::list и std::forward\_list, std::set и std::map, std::unordered\_set и std::unordered\_map? В чём фундаментальная разница между устройством этих контейнеров?

List – двусвязанные отдельные ячейки памяти (можно бегать в обе стороны)

forward\_list – односвязный список, как лист, только в одну сторону

Set – дерево, скорее всего, тоже отдельные ячейки памяти (не соседние)

Map – структура похожая на set, только каждому элементу ставится в соответствие ещё один

std::unordered\_set и std::unordered\_map – создано как таблицы.

1. (5 б.) Какие математические объекты могут быть сопоставлены контейнерам std::set и std::map?

Множество и функция (словарь)

1. (5 б.) Почему при инициализации std::array необходимо задавать размер?

Потому что память выделяется в стеке, следовательно, нужно знать размер массива.