

Контрольные вопросы:

1. (8 б.) Что такое стек и куча в контексте модели памяти C++? Какие переменные хранятся на стеке / в куче?
2. (8 б.) Каким образом связаны встроенные указатели и массивы?
3. (8 б.) Почему низкоуровневая работа с памятью небезопасна?
4. (8 б.) Какие конструкции языка C++ аналогичны следующим концепциям Python: `list`, `list.append()`, `list.clear()`, `for item in some_list`, `str`?
5. (8 б.) Для чего нужен оператор `delete`? В чём разница между операторами `delete` и `delete[]`?

Упражнения:

1. (25 б.) Тензор второго ранга — это линейный оператор, переводящий векторное пространство в себя. Если в пространстве задан базис, то тензор может быть представлен в виде квадратной матрицы. Введите с клавиатуры вектор и матричное представление тензора в некотором базисе и воздействуйте тензором на вектор. Выведите результат. Реализуйте решение для вектора размерности 3 или 4 по вашему выбору. (Соответственно, представление тензора — это матрица размеров 3x3 или 4x4).
2. (25 б.) Реализуйте функцию `split()` — аналог `str.split()` в Python. Она должна принимать строку и возвращать вектор строк, разбитых по разделителю.
Подсказка: воспользуйтесь методами строки `find()` и `substr()`. Посмотрите примеры в документации!
3. (25 б.) Реализуйте алгоритм сортировки вставками массива чисел. Потренируйтесь использовать контейнер `std::vector` без итераторов и функцию обмена `std::swap`. Псевдокод алгоритма можете посмотреть [здесь](#).
4. (25 б.) Реализуйте алгоритм бинарного поиска в отсортированном массиве чисел. Достаточно установить факт наличия или отсутствия искомого числа в массиве. Потренируйтесь использовать контейнер `std::vector` без итераторов и оптимизируйте цикл поиска. Псевдокод алгоритма можете посмотреть [здесь](#).