

1. Реализуйте функцию `swar_min`, которая принимает на вход двумерный массив целых чисел, ищет в этом массиве строку, содержащую наименьшее среди всех элементов массива значение, и меняет эту строку местами с первой строкой массива. Подумайте, как обменять строки массива, не обменивая элементы строк по-отдельности.
2. Реализуйте функцию `getline`, которая считывает поток ввода посимвольно, пока не достигнет конца потока или не встретит символ переноса строки (`'\n'`), и возвращает C-style строку с прочитанными символами.  
Обратите внимание, что так как размер ввода заранее неизвестен, то вам нужно будет перевыделять память в процессе чтения, если в потоке ввода оказалось больше символов, чем вы ожидали.  
Память, возвращенная из функции будет освобождена оператором `delete[]`. Символ переноса строки (`'\n'`) добавлять в строку не нужно, но не забудьте, что в конце C-style строки должен быть завершающий нулевой символ.
3. В этой задаче вам нужно реализовать функцию `resize`. Функция `resize` принимает на вход три аргумента: указатель на область памяти, выделенную с помощью оператора `new[]`, размер области (`size`) и новый размер (`new_size`). Функция должна выделить память размера `new_size`, скопировать в нее данные из переданной области памяти, освободить старую область памяти и вернуть выделенную область памяти нового размера со скопированными данными.  
Обратите внимание, что память, возвращенная из функции, будет освобождена с помощью оператора `delete[]`. Несмотря на то, что входной тип данных — это `const char *`, не стоит рассчитывать, что там хранится C-style строка. При выполнении задания также учтите, что `new_size` может быть меньше `size`.
4. К заданию должна быть написана обертка, тестирующая ваши функции (можно несколько разных оберток). Программа должна запускаться под `valgrind` с проверкой на утечки памяти. Утечек памяти быть не должно ;-)