

Тексты всех трех задач расположены в соответствующих каталогах, в файлах с именем Task.txt. Файлы во всех трех каталогах устроены так же, как и файлы, которые будут доступны вам на коллоквиуме (тесте), с той разницей, что папки с упражнениями содержат еще и решение (которое, естественно, не будет доступно вы на коллоквиуме):

1. Файл *задача.txt* содержит текст задания, а также пример того, как должно выглядеть взаимодействие пользователя с программой.
2. Файл *test.sh* содержит скрипт, используемый для тестирования программы. Вы запускаете сценарий, помещая себя в каталог, где находится сценарий, а также ваше решение (файл *.S*), а затем в терминале вы вводите *./test.sh my\_solution.S*. Таким образом, вы передаете скрипту файл кода (*.S*), а не файл, который вы получили в результате компиляции. Скрипт скомпилирует программу и запустит ее с конкретными тестовыми примерами. Существует вероятность того, что изначально у вас не будет прав на запуск этого сценария. В этом случае измените привилегии одним из следующих способов:

а. Щелкните правой кнопкой мыши на *test.sh* → *характеристики* → открыть вкладку *Разрешения* → поставь галочку *Разрешить выполнение файла как программы*

б. Откройте терминал в каталоге, где он находится. *test.sh* скрипт → тип:

`chmod +x test.sh`

После запуска теста указанным способом вы получите для нескольких тестовых примеров информацию о том, какие входные данные были переданы в вашу программу, какие выходные данные являются ожидаемыми, а также какой результат выдала ваша программа. В конце также будет написано, сколько тестов было пройдено успешно.

**Примечание:** тест, вероятность успеха которого составляет 100%, т.е. который проходит все тестовые примеры, **это не гарантия того, что ваша программа и ваше решение полностью верны**. Вам следует позаботиться о том, чтобы программа охватывала пограничные случаи (*крайние случаи*). Рекомендация — сначала написать программу, которую вы будете тестировать самостоятельно, без использования скрипта, на собственных примерах и отладчике, и только потом приступить к автоматическому тестированию с помощью скрипта.

3. Файл с подготовленными переменными и строками, в которые следует записать решение. В каждой задаче этот файл имеет разное имя. Каждый такой файл, а также тот, который будет доступен вам на коллоквиуме, заранее содержит определенное количество переменных и определенное количество строк. Это важно **не меняйте имена переменных и содержимое строк**. Если вы измените какой-либо из подготовленных предметов, **ни один тестовый пример не пройдет**. Эти переменные и строки важны, поскольку тестовый сценарий будет ожидать их во время выполнения точно в том виде, в каком они есть в файле заранее. Вы можете добавлять строки кода до, после и между строками уже существующих строк и переменных, и это не повлияет на выполнение, но не меняет их содержимое само по себе. Вы записываете свое решение в этот файл и передаете его тестовому сценарию.

Остальная часть документа содержит примечания, относящиеся к поставленным задачам, а также некоторые дополнения:

## Задание 1:

- Протестировать эту задачу с помощью скрипта можно только после того, как вы выполнили часть б). Если вы не выполните эту часть, тесты не пройдут. Сменный *имепрез* Он предназначен для ввода с клавиатуры с максимальной длиной 50 символов (включая *входить*).

## Задача 2:

- Основная задача должна проверить **полное совпадение символов**с обеих сторон строны. Итак, если у вас есть строка *Ана любит миловану*, это будет палиндром, но строка *Ана*этого не произойдет, потому что с одной стороны большая буква «А», а с другой — маленькая. Кроме того, если строка *Ана любит миловану*,это не будет палиндром, потому что четвертый символ слева — это пробел, а четвертый справа — буква «у». Тестовый сценарий проверит это решение. Сменный *вход*должен принимать ввод с клавиатуры длиной не более 50 символов (включая *входить*).
- Приложение 1:**Измените первую задачу так, чтобы она не различала прописные и строчные буквы (так *Ана*будет палиндром).
- Приложение 2:**При проверке игнорируйте все, что не является символом (пробелы, специальные символы, цифры; *Ана любит миловану*является палиндромом; *Ана любит миловану*это палиндром).
- Решение *palindrome\_resenje.S*— решение основной задачи, без дополнений.
- Решение *palinrdom\_prosireno\_resenje.S*— решение задачи с обоими плагинами. В этом решении основное внимание уделяется самому алгоритму, поэтому нет ввода с клавиатуры и печати на экране, но строка в начале находится в памяти. Кроме того, код выхода помещается в переменную, а не в возвращаемое значение программы.
- Запишите свое решение в подготовленный файл *.палиндром.S*.
- Файл *палиндромы-srp.txt*содержит список палиндромов, которые можно использовать для ручного тестирования (обратите внимание, что они содержат пробелы).

## Задача 3:

- Приложение:**Выведите введенные строки в лексикографическом порядке. Сначала напишите тот, первая буква которого лексикографически наименьшая (ближайшая к началу алфавита). Если две строки начинаются с одной и той же буквы, редактируйте их по тому же принципу, исходя из второго символа. Если остальные символы такие же, обратите внимание на третьего и так далее.
- Решение *stringlist\_resenje.S*— решение основной задачи, без дополнений.
- Решение *stringlist\_resenje\_dodatak\_sortinarje.S*это решение проблемы с дополнением.
- Запишите свое решение в подготовленный файл *.список строк.S*. Дополнительно определите необходимые переменные согласно инструкциям из текста задачи (файл *задача.txt*).

## Дополнительное упражнение:

- Напишите программу на ассемблере, которая объединяет две строки (входная строка имеет длину до 50 символов, выходная — 100). Взаимодействие с программой должно выглядеть так:

Введите первую строку: первая

Введите вторую строку: вторая

Объединение: первая секунда

- Для этой задачи нет подготовленных файлов или тестов.