**1. Калькулятор среднего значения**

Описание: Создайте скрипт, который находит среднее значение из предоставленных чисел.

Требования:

- Скрипт принимает набор чисел в качестве аргументов командной строки.

- Используйте массив для хранения чисел.

- Реализуйте функцию для вычисления среднего значения.

**2. Конвертер температур**

Описание: Создайте bash-скрипт для преобразования температур из градусов Цельсия в градусы Фаренгейта и наоборот.

Требования:

- Скрипт должен хранить массив температур

- Реализуйте функции для преобразования из Цельсия в Фаренгейт и из Фаренгейта в Цельсий.

- Пользователь может добавить новую температуру в массив

- Пользователь может выбрать как посмотреть все температуры: в Цельсиях или Франгейтах

**3 . Проверка палиндромов**

Описание: Напишите скрипт для проверки, являются ли заданные слова палиндромами.

Требования:

- Скрипт в аргументах произвольное количество слов.

- Реализуйте функцию для проверки палиндрома.

1. Проверьте каждое слово, является ли оно палиндромом.

2. Выведите результат для каждого слова.

**4. Сортировка чисел**

Описание: Напишите скрипт для сортировки заданного пользователем набора чисел.

Требования:

- Скрипт принимает числа в качестве аргументов командной строки.

- Реализуйте функцию для сортировки массива, отсортируйте массив и выведите результат на экран.

**5. Генерация паролей**

Описание: Создайте скрипт для генерации указанных пользователем случайных паролей.

Требования:

- Скрипт принимает в аргументах командной строки количество паролей, которые надо сгенерировать, и их длину.

- Реализуйте функцию для генерации пароля заданной длины.

- С помощью функции сгенерируйте пароли и выведите их на экран.  
  
Можно использовать, например, /dev/urandom: это специальный файл в UNIX-системах, который предоставляет случайные данные

**6. Проверка уникальности**

Описание: Напишите скрипт для проверки, все ли элементы в заданном массиве уникальны.

Требования:

- Скрипт принимает элементы массива в виде аргументов командной строки.

- Реализуйте функцию для проверки уникальности элементов массива.

- Выведите на экран сообщение, есть ли в массиве дубликаты или нет.

**7. Автоматическое переименование файлов:**

Описание: Разработайте скрипт для автоматического переименования всех файлов в указанной директории.   
  
- Скрипт принимает в аргументах командной строки новый суффикс для файлов и папку, в которой нужно переименовать файлы;

- Скрипт переименовывает все файлы в указанной папке по шаблону:  
имя\_суффикс.расширение

- Переименование реализовать отдельной функцией

Пример:  
  
./script old dirName  
  
В папке dirName 2 файла: test.txt и music.mp3

После выполнения скрипта имена файлов станут такими: test\_old.txt и music\_old.mp3

**8. Сжатие и архивирование файлов:**

Описание: Напишите скрипт, который сжимает и архивирует файлы из указанной директории, чьи размеры превышают заданный порог.

Требования:

- Скрипт принимает в аргументах командной строки папку, в которой нужно архивировать файлы, и максимальный размер файла;

- Архивирование реализовать в отдельной функции;

Для архивирования можно использовать команду tar:

tar -cf archive.tar filename  
  
archive.tar – ваше название файла-архива

filename – имя файла, которого архивируете  
  
Например, в задаче можно так tar -cf filename.tar filename

**9.** **Создание и управление архивами логов:**

Описание: Напишите скрипт, который управляет архивами в указанной директории.

Требования:

- Скрипт принимает в аргументах командной строки папку, в которой нужно архивировать файлы, а так же режим работы:

- Режимы работы:  
1. Архивировать все файлы в папке с добавлением текущей даты к названию файла;

2. Удалить все файлы, старше заданной даты.

Для архивирования можно использовать команду tar:

tar -cf archive.tar filename  
  
archive.tar – ваше название файла-архива

filename – имя файла, которого архивируете  
  
Например, в задаче можно так tar -cf filename.tar\_date filename

Для 2 режима дату из названия файла можно вырезать с помощью "${file##\*\_}"

**10. Мониторинг использования дискового пространства:**

Описание: Напишите скрипт, который проверяет использование дискового пространства на указанном пользователем разделе диска и выдает предупреждение, если использование превышает заданный процент.

Требования:

- Директорию и процент указывать через аргументы командной строки;

- Проверку реализовать в отдельной функции

Chatgpt:

#!/bin/bash

# Функция для проверки использования дискового пространства

check\_disk\_usage() {

local directory=$1

local threshold=$2

# Получаем использование дискового пространства в процентах для указанного раздела

usage=$(df -h "$directory" | awk 'NR==2 {print $5}' | sed 's/%//')

# Проверяем, превышает ли использование заданный порог

if [ "$usage" -gt "$threshold" ]; then

echo "Предупреждение: использование диска в '$directory' составляет $usage% и превышает установленный порог в $threshold%"

else

echo "Использование диска в '$directory' в пределах нормы: $usage%"

fi

}

# Проверка наличия необходимых аргументов

if [ $# -ne 2 ]; then

echo "Использование: $0 <directory> <threshold>"

exit 1

fi

# Вызываем функцию с аргументами командной строки

check\_disk\_usage "$1" "$2"

**11. Управление списком задач:**

Описание: Напишите скрипт для работы со списком задач, который позволяет добавлять новые задачи и отображать текущий список задач, хранящийся в текстовом файле.

Требования:

- Имя файла с задачами передавать через аргументы командной строки;

- Сделать 2 функции: добавление задач в файл и просмотр всех задач.

**12. Система учета расходов:**

Описание: Создайте скрипт, который позволяет пользователю добавлять и просматривать записи о расходах.

Требования:

- Скрипт должен сохранять данные в текстовом файле

- Имя файла передавать в аргументах командной строки

- Добавление новой записи реализовать в отдельной функции. Каждая запись должна включать дату, категорию расхода, сумму и описание.

**13. Модуль управления контактами:**

Описание: Разработайте скрипт для управления контактной книгой в файле.

Требования:

- Скрипт должен позволять пользователям добавлять и удалять контакты.

- Каждый контакт должен включать имя, телефон и email.

- Контакты должны храниться в текстовом файле, и скрипт должен предоставлять возможность поиска по имени или номеру телефона. (например, использовать grep)

- Поиск и добавление реализовать в отдельных функциях

**14. Фильтрация дубликатов:**

- Создайте скрипт, который принимает массив строк, находит и удаляет из него дублирующиеся элементы.

Требования:

- Массив строк передавать в аргументах командной строки;

- Фильтрацию сделать в отдельной функции;

- Выведите обновлённый массив, содержащий только уникальные строки.