

1. Написать скрипт определяющий запущен ли скрипт пользователем root.
2. Написать скрипт который после запуска ожидает ввод строки от пользователя, а затем выводит ее в стандартный вывод. Используйте команду read.
3. Написать скрипт который после запуска ожидает ввод строки от пользователя, а затем выводит ее в стандартный вывод, но ограничить время ожидания пользовательского ввода 5 секундами. В случае если пользователь не успел ввести строку, остановить программу и напечатать в стандартный вывод сообщение об этом.
4. Напишите скрипт, возвращающий
  - 4.1. количество дней, прошедших с начала года
  - 4.2. количество секунд, прошедших с начала "эпохи UNIX".
5. Написать скрипт, который выводит в стандартный вывод все аргументы который были ему переданы. Используйте команду shift.
6. Напишите скрипт который ожидает пользовательского ввода одного символа с клавиатуры, проверяет что символ был введен один. Далее скрипт определяет что за символ был введен - буква в нижнем регистре, буква в верхнем регистре, цифра, знак пунктуации, пробел или что-то другое. Используйте команду case.
7. Написать скрипт преобразующий метры в мили. В качестве входящего аргумента должна быть цифра - метры. В стандартный вывод вывести количество миль.
8. Напишите скрипт который будет проводить симуляцию 700 бросков 6 гранного кубика. Вывод должен быть в следующем формате:

```
echo "единиц    =    $ones"
echo "двоек     =    $twos"
echo "троек     =    $threes"
echo "четверок  =    $fours"
echo "пятерок   =    $fives"
echo "шестерок  =    $sixes"
```

1.2.1. Написать скрипт, который рандомно сгенерирует на диске структуру файлов и папок. Входным параметром будут: директория, в которой нужно генерировать данные, глубину вложенности поддиректорий, максимальный размер файлов, которые будут создаваться в этих директориях и максимальное количество итераций по созданию объектов в директории (т.е. общее количество файлов и директорий, которые будут созданы). Максимальная длина имени - 8 символов, содержимое файлов - рандом или нули. Проверку на доступное место делать не нужно.

1.2.2. Написать скрипт, который принимает 2 входных параметра: source\_dir и destination\_dir - две директории на диске, сравнивает, что они ни одна из них не является родительской для другой, у них не совпадают имена и т.п. Далее скрипт должен подсчитать место, занимаемое source\_dir, и место, доступное на диске, где находится директория destination\_dir, если есть доступное место - скопировать (с наследованием всех атрибутов: времени создания, доступа и модификации, владельца и т.п.), если места нет - вывести предупреждение на экран и предложить пользователю продолжить (С или Y) или прервать (N или A).

1.2.3. Модернизировать скрипт, чтобы он упаковывал данные из source\_dir и складывал в destination\_dir. При этом в самом начале работы скрипта нужно предусмотреть диалог с пользователем: предложить дописывать текущую дату и время в формате YYMMDD\_HHSS.gz к имени файла или выполнять ротацию архивов, т.е. дописывать порядковый номер архива (0.gz, 1.gz, 2.gz). В случае выбора второго варианта - так же нужно запросить максимальное количество копий, при этом номер 0 присваивается самому последнему созданному архиву, а номер N-1 - самому старому, на каждой итерации архив с номером N - удаляется.

1.2.4. Модернизировать скрипт, чтобы все сообщения об ошибках подавлялись, а стандартные - выводились на консоль + записывались в лог-файл в текущей директории с именем out\_YYMMDD\_HHSS.log. Т.к. сообщения об ошибках подавляются, то в случае возникновения ошибок в процессе работы скрипта (проблемы с пермишенами, нехватка места, отсутствие какой-либо утилиты и т.п.) необходимо в конце работы скрипта вывести сообщение в консоль красным цветом

"Warning: X error(s) occurred!", где X - количество перехваченных ошибок или количество вызовов команд, которые повлекли ошибки (как удобнее, на выбор).

1.3.1. Написать скрипт, который бы выводил на экран наиболее выгодный курс валют конвертации USD->BYR и EUR->BYR. Способ работы скрипта: либо передавать входным параметром "USD" или "EUR" и выводить курс, либо, если входных параметров нет - а консольном диалоге запросить на выбор желаемый курс. Получать можно с любого сайта (в идеале - с [finance.tut.by](http://finance.tut.by)).

1.3.2. Написать скрипт, который мог бы пройти авторизацию на Яндексe, сохранить сессию в cookie-файл, и показать кол-во писем в папке Inbox.