Отчёт по выполнению 3 этапа внешнего курса

Операционные системы

Бережной Иван Александрович

Содержание

# 1 Цель работы

Научиться работать с операционной системой Linux

# 2 Задание

Выполнить тестовые задания на основе теории

# 3 Выполнение работы

Выполняем третий этап: (рис. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37).

Проверим, какой из вариантов сработает в редакторе и выберем его.

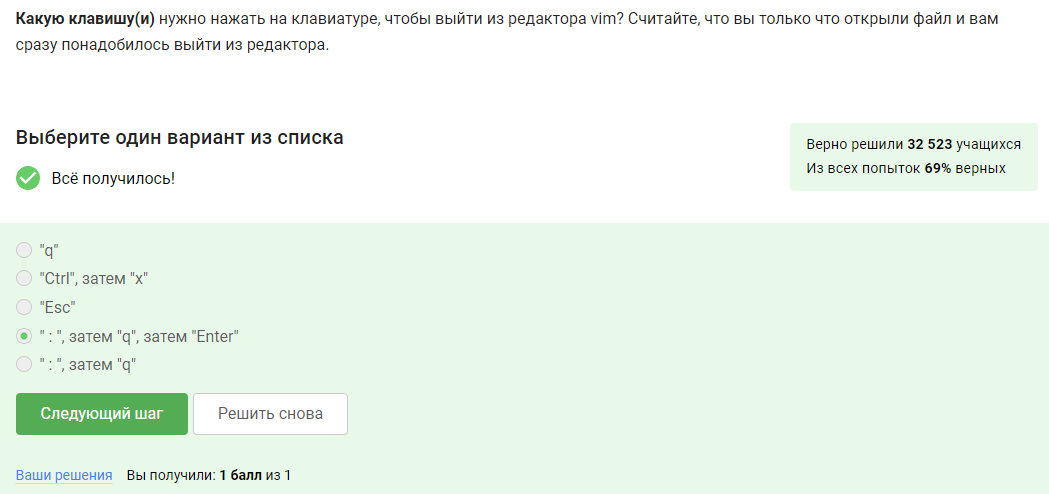


Рис. 1: Задание 1

Создаём текстовый файл vim tmp.txt, копируем туда строку “Strange\_ TEXT is\_here. 2=2 YES!” и проверяем каждый вариант.

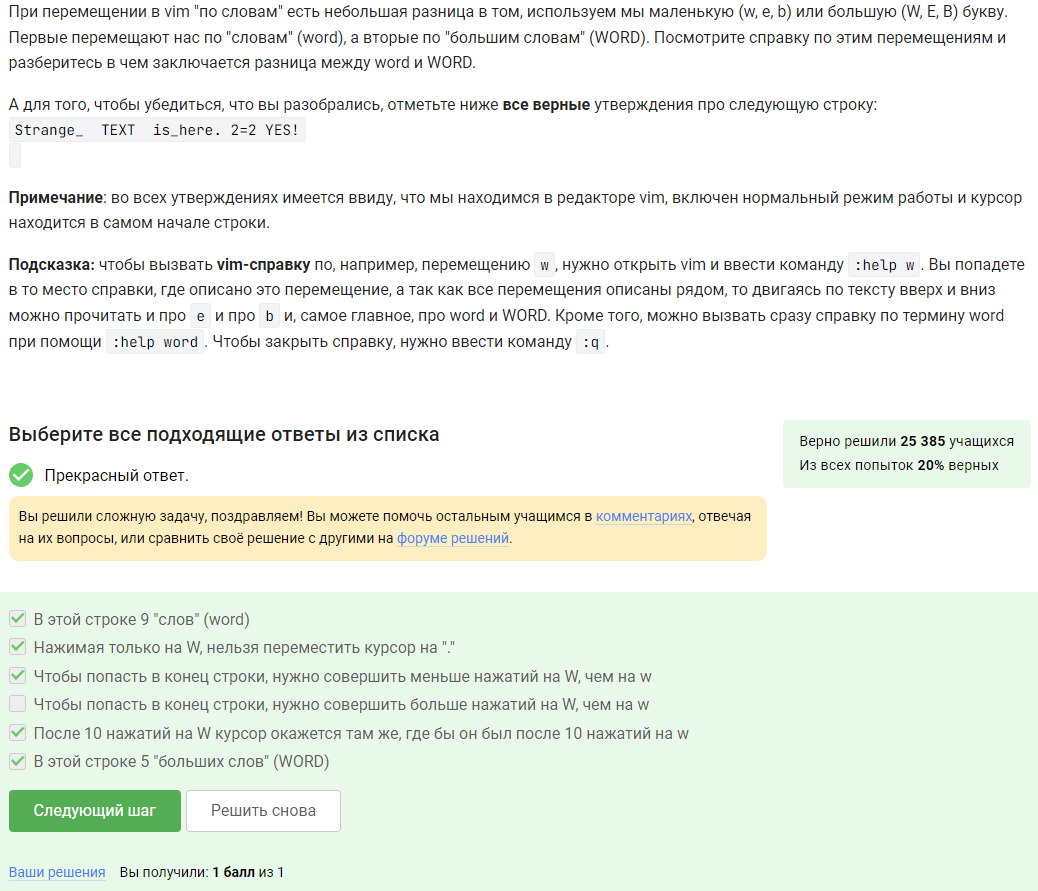


Рис. 2: Задание 2

Проверим каждый набор клавиш и выберем верные варианты ответа.

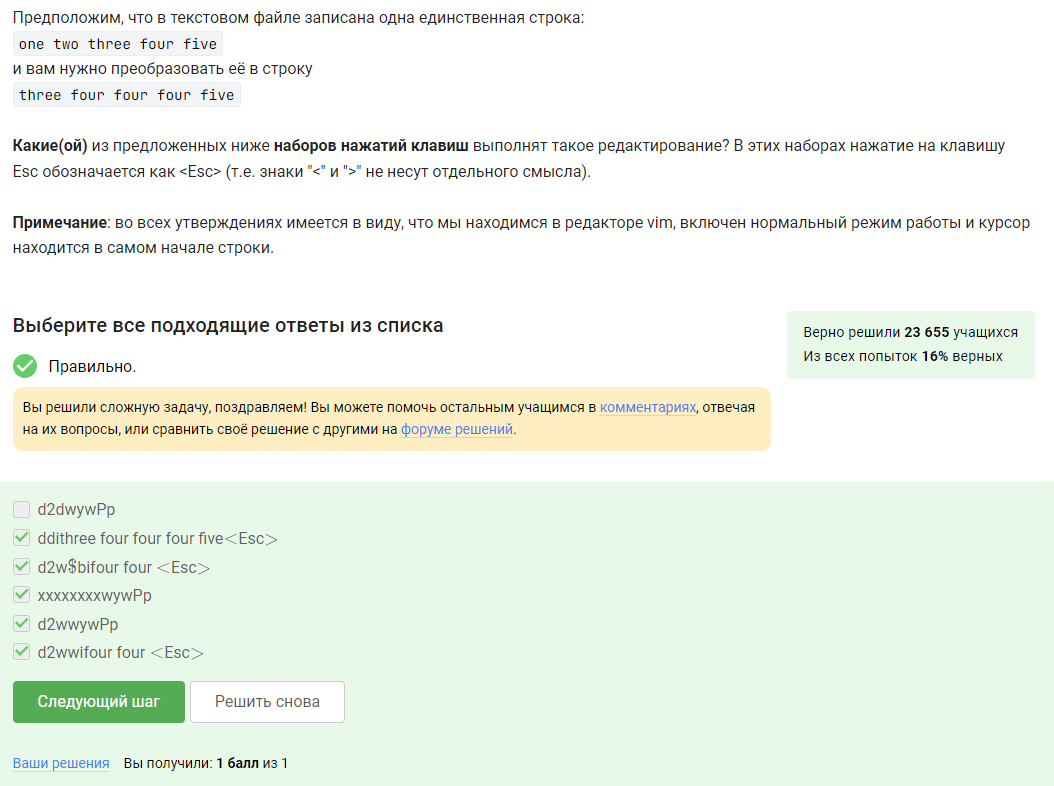


Рис. 3: Задание 3

Поиск и замена в Vim осуществляется командой :substitute, однако куда удобнее использовать для нее аббревиатуру :s. Общий синтаксис этой команды примерно такой:

:{пределы}s/{что заменяем}/{на что заменяем}/{опции}

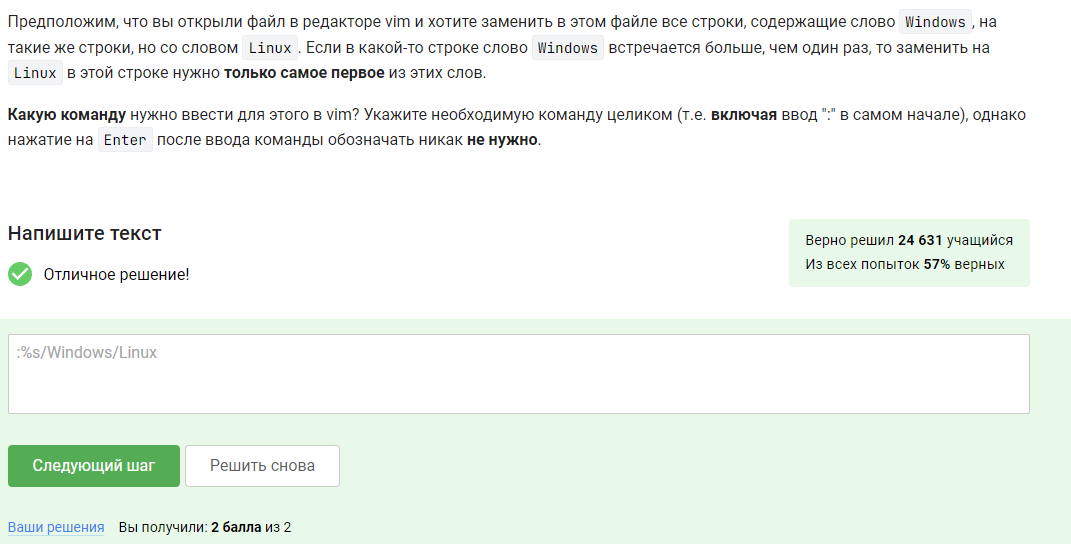


Рис. 4: Задание 4

Опять же, на практике проверим каждое утверждение.

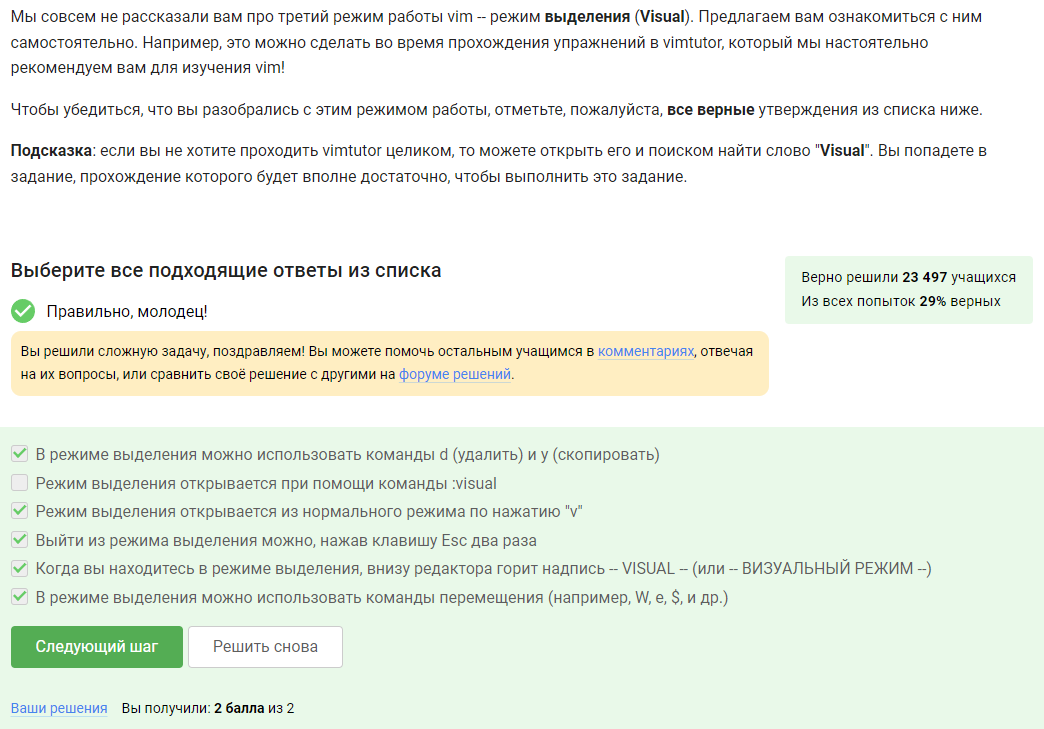


Рис. 5: Задание 5

Только из набора С потому что у каждой оболочки свой буфер, который при выходе из нее буде записываться в файл истории.

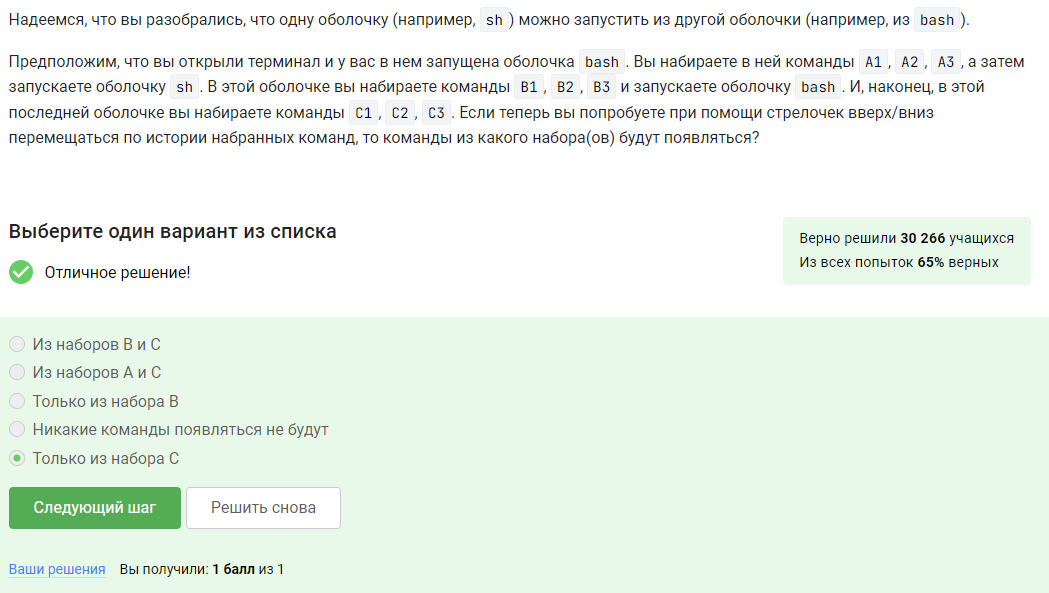


Рис. 6: Задание 6

/home/bi/file1.txt - потому что именно в этой директории мы создаем новый файл, а уже после его создания мы переходим в другую папку.

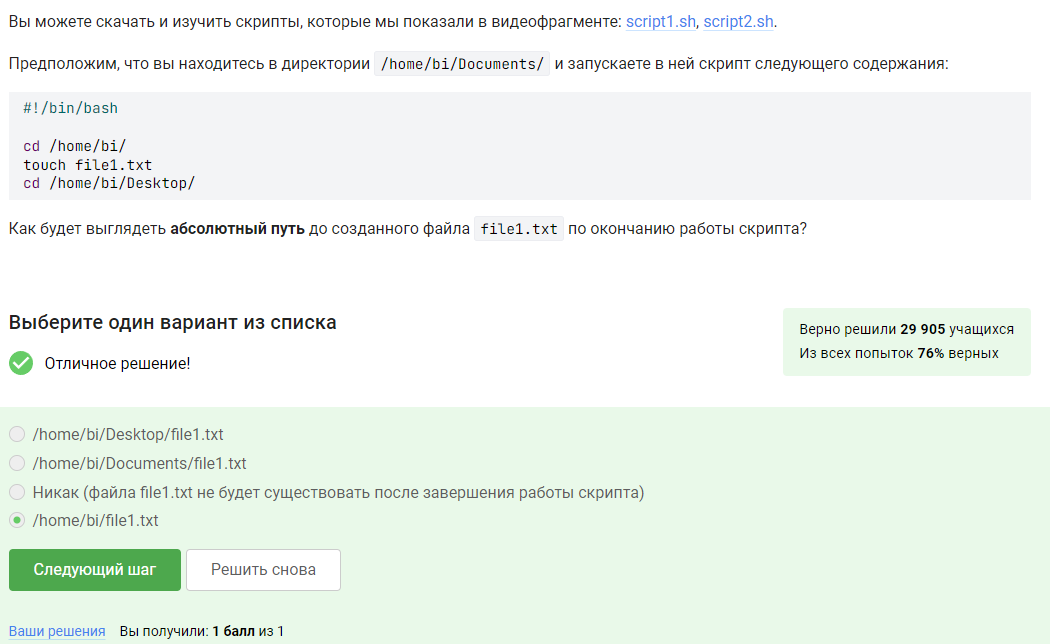


Рис. 7: Задание 7

В имени могут быть только буквы, цифры и подчеркивание.

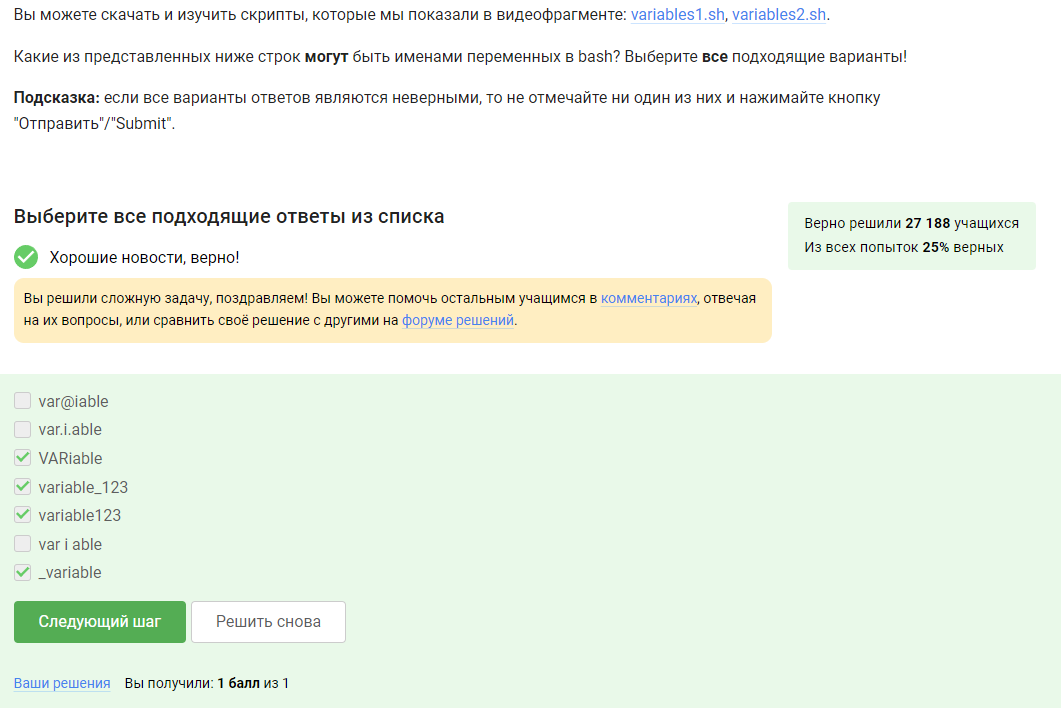


Рис. 8: Задание 8

$ echo опции строка Эта команда печатает строки, которые передаются в качестве аргументов в стандартный вывод и обычно используется в сценариях оболочки для отображения сообщения или вывода результатов других команд.  
var1=$1 - обозначение переменных  
var2=$2` "Arguments are: \$1=$var1 $2=$var2” - строка печати.

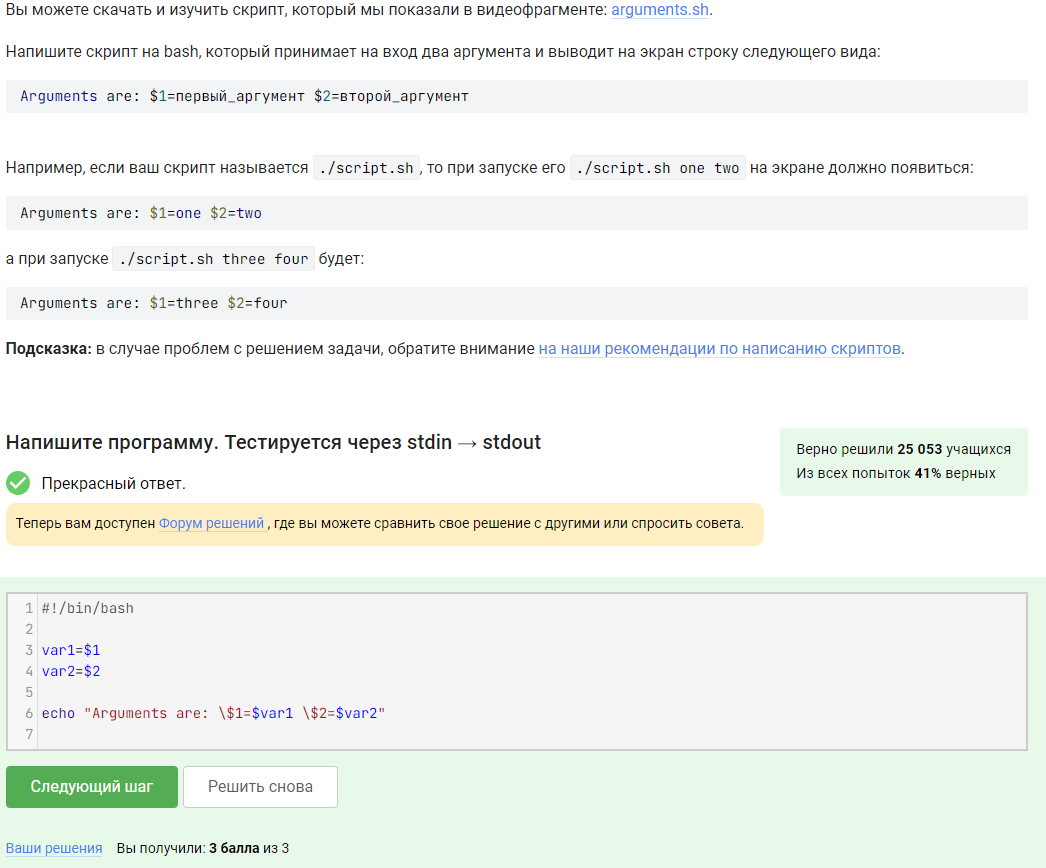


Рис. 9: Задание 9

$0 #имя скрипта  
-ge, # больше или равно  
-gt, (>) # больше  
-s <path> # разм файла больше 0 -le, # меньше или равно

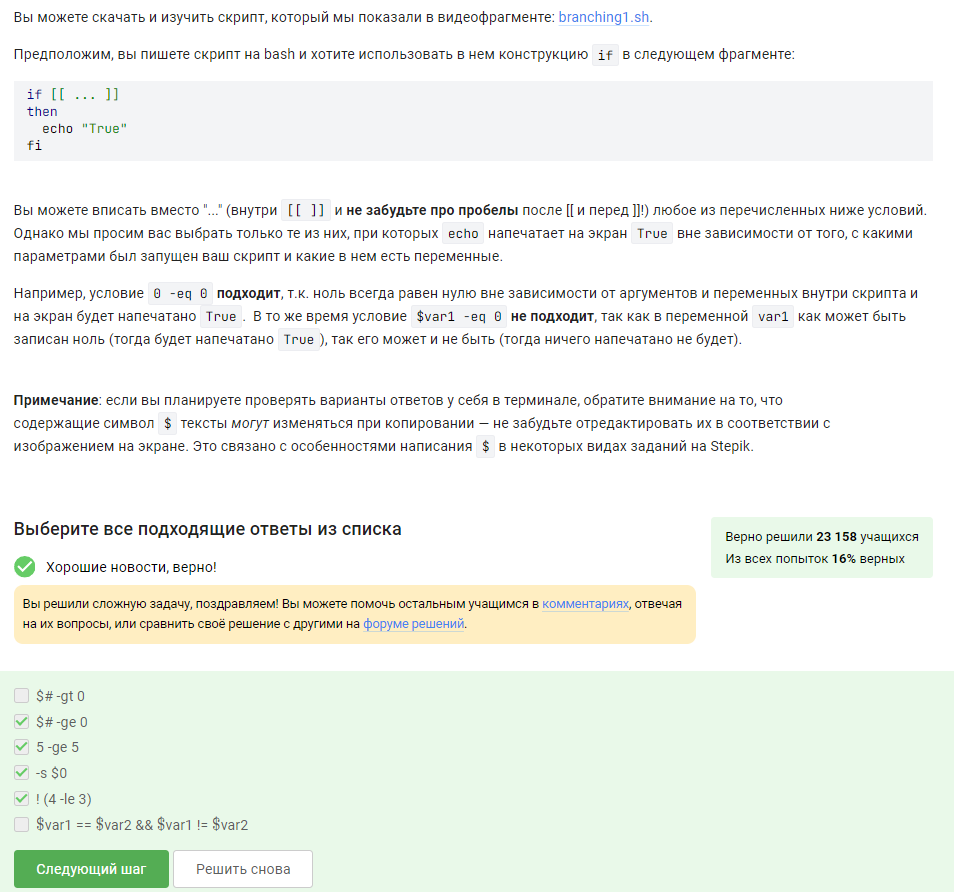


Рис. 10: Задание 10

1. Задаю общую часть в каждом выводе - слово “student”: v=student
2. Выполняем команды для разных аргументов.
3. res - это результат для вывода
4. echo “$res” - вывести результат

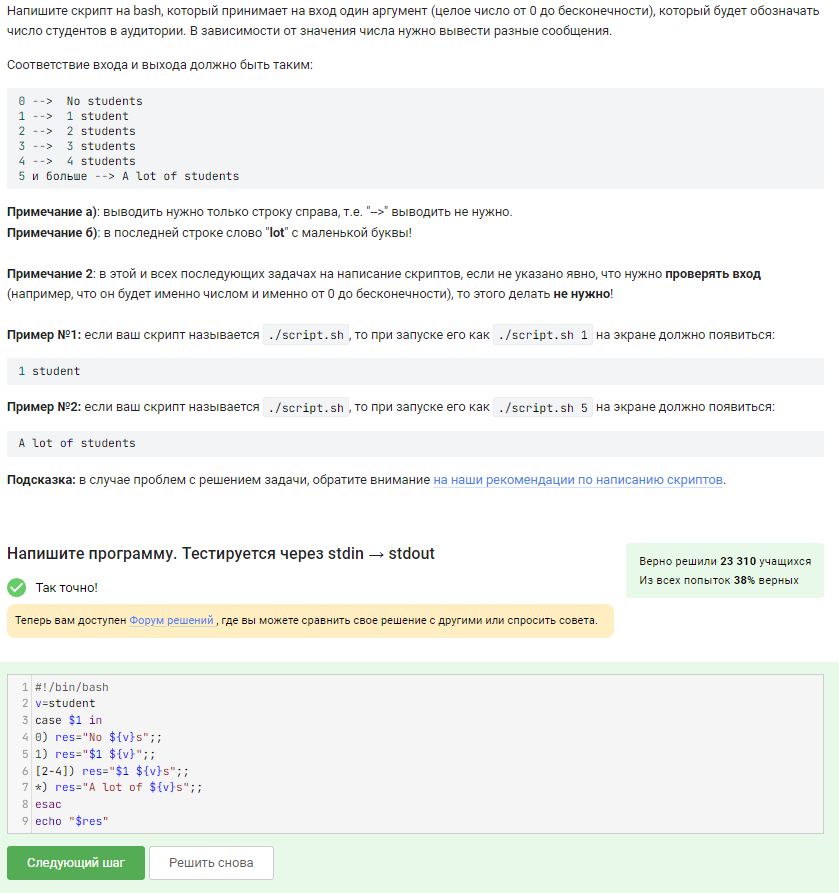


Рис. 11: Задание 11

a > c нет (Start, Finish)

, > c нет (Start, Finish)

b > c нет (Start, Finish)

c\_d > c да (Start)

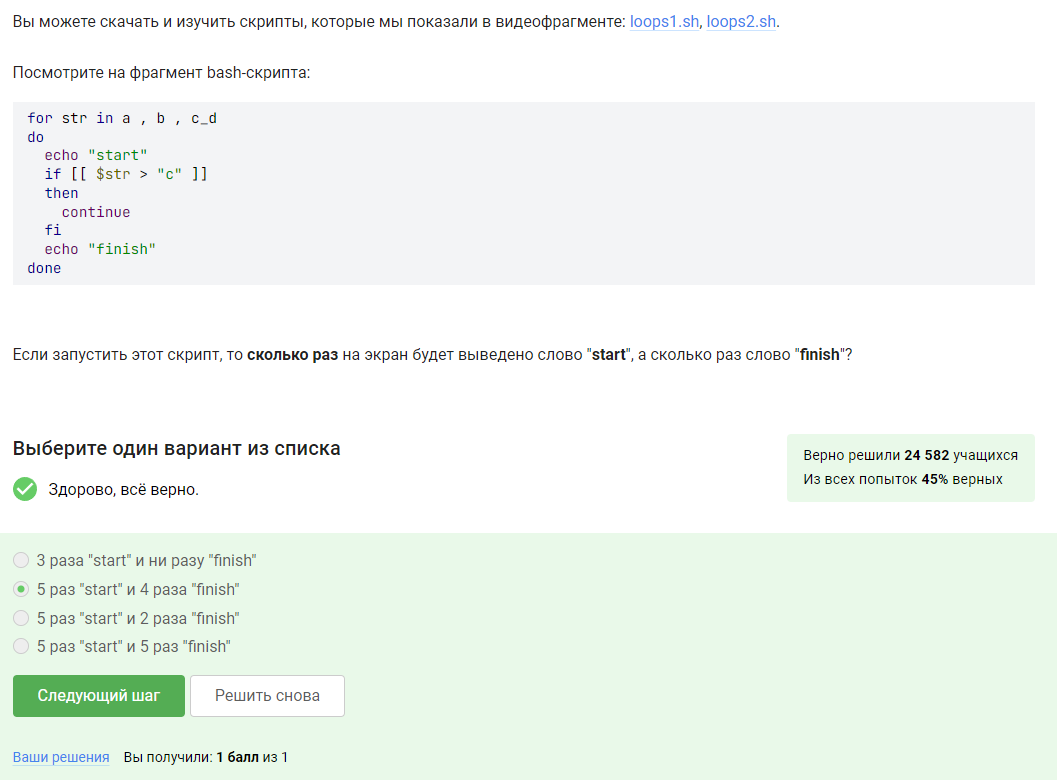


Рис. 12: Задание 12



Рис. 13: Условие задания 13

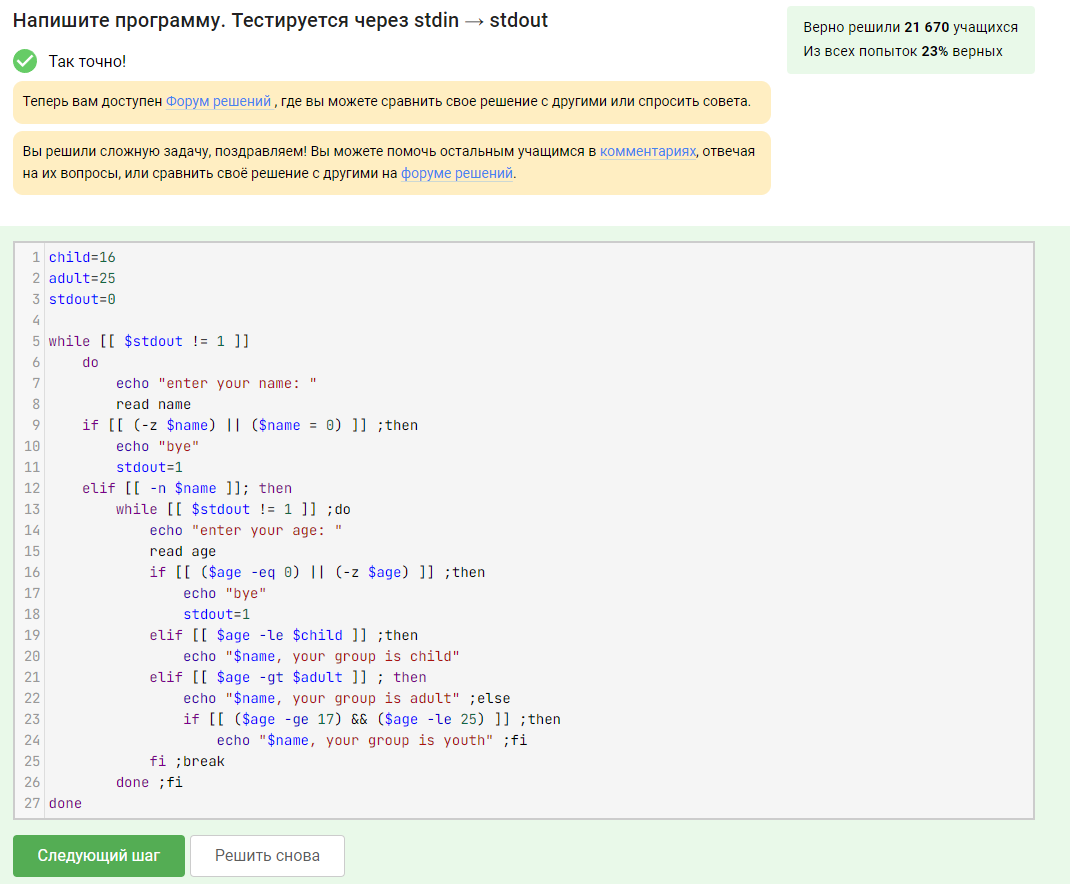


Рис. 14: Задание 13

1. a = $a
2. a += b это то же самое, что и a = a + b, но с символами “+=” != “=+”
3. если выражение не в скобках, но с пробелами - работать не будет. (let a=a+b - сработает; let a = a + b - нет)

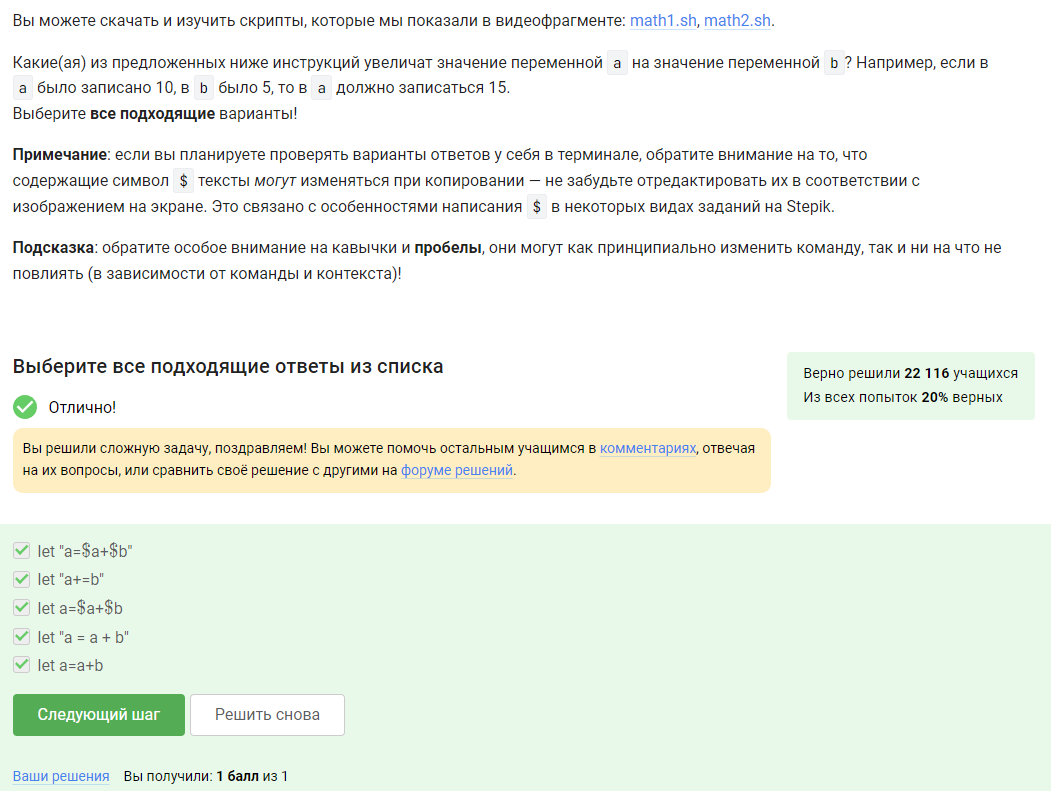


Рис. 15: Задание 14

programm в condition выполняет стандартный вывод в терминал. Если программа не должна выводить в терминал rm cp etc… то она возвращает true || false. Нужно перенаправить вывод (для тех, которые выводят) и programm возвращает true || false как в случае с теми прогами которые не выводят.

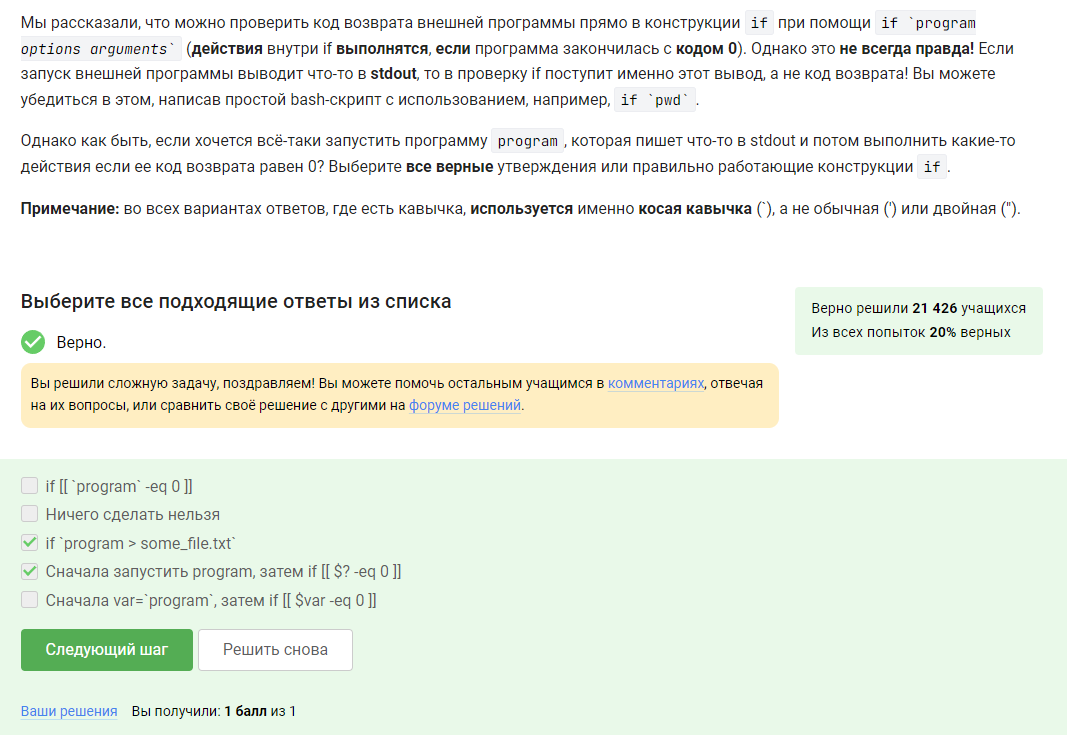


Рис. 16: Задание 15

Первая переменная локальная, и это просто пустая строка, вторая переменная - это сумма арифметической прогрессии от 1 до 10, равна 55, но при умножении на 2 даст 110.

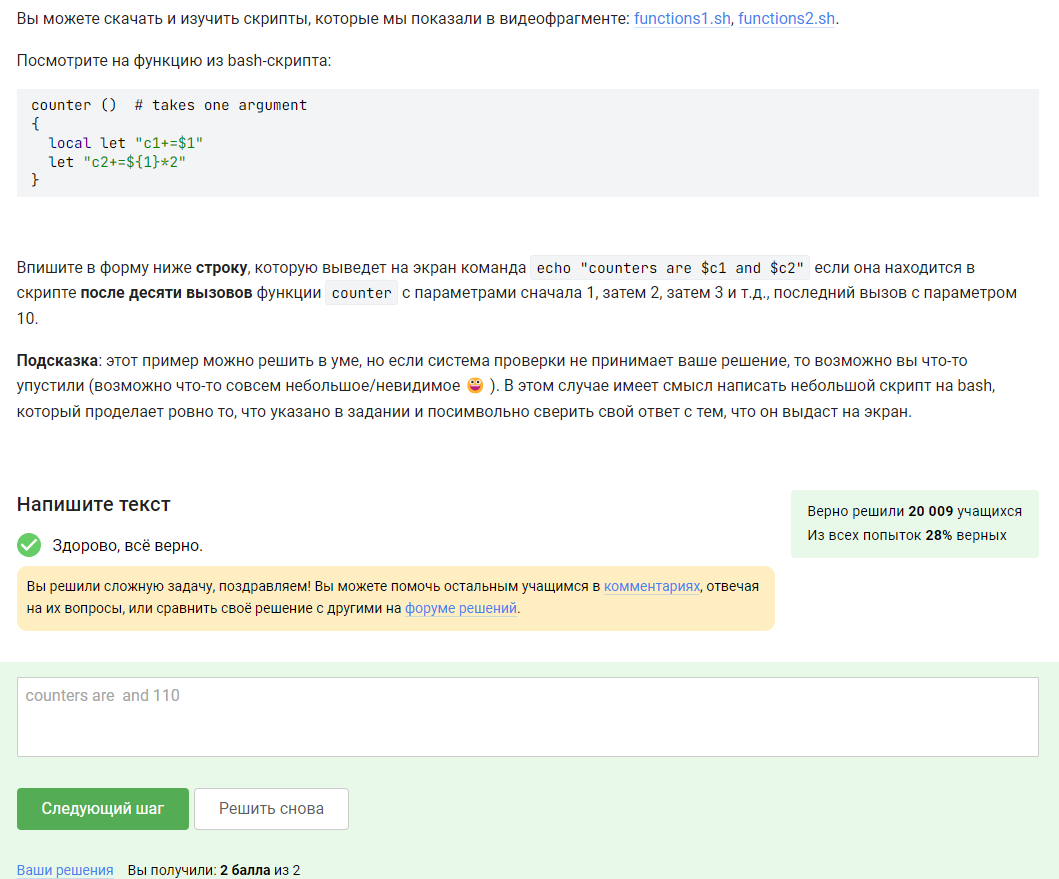


Рис. 17: Задание 16

Алгоритм нахождения НОД делением 1. Большее число делим на меньшее. 2. Если делится без остатка, то меньшее число и есть НОД (следует выйти из цикла). 3. Если есть остаток, то большее число заменяем на остаток от деления. 4. Переходим к пункту 1

|  |
| --- |
| Условие задания 17 |

Рис. 18: Условие задания 17

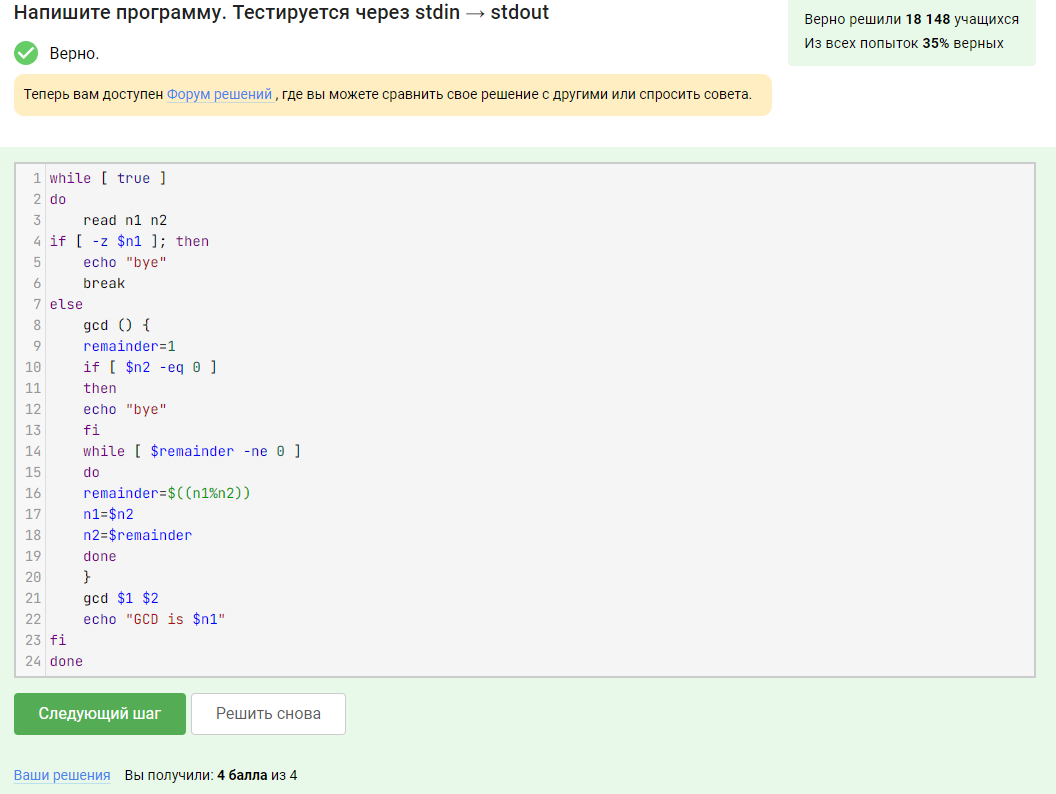


Рис. 19: Задание 17

Калькулятор выглядит обычно - мы вводим два числа, пишем, что с ними надо сделать, и потом, учитывая случаи ошибок, выводим результат.

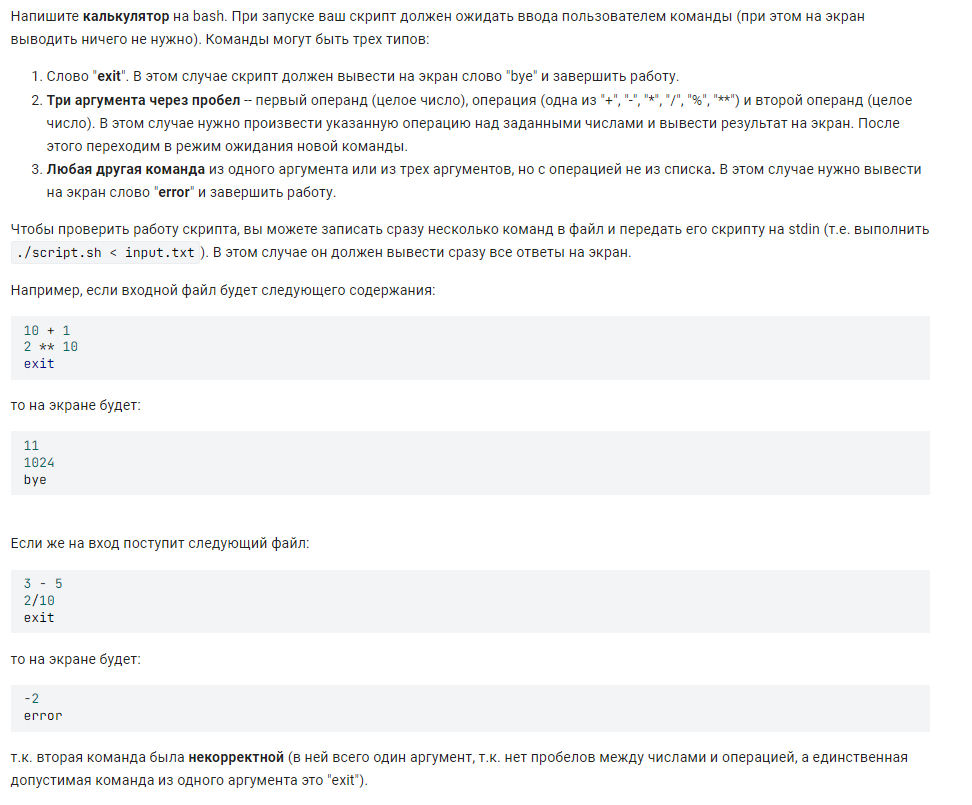


Рис. 20: Условие задания 18

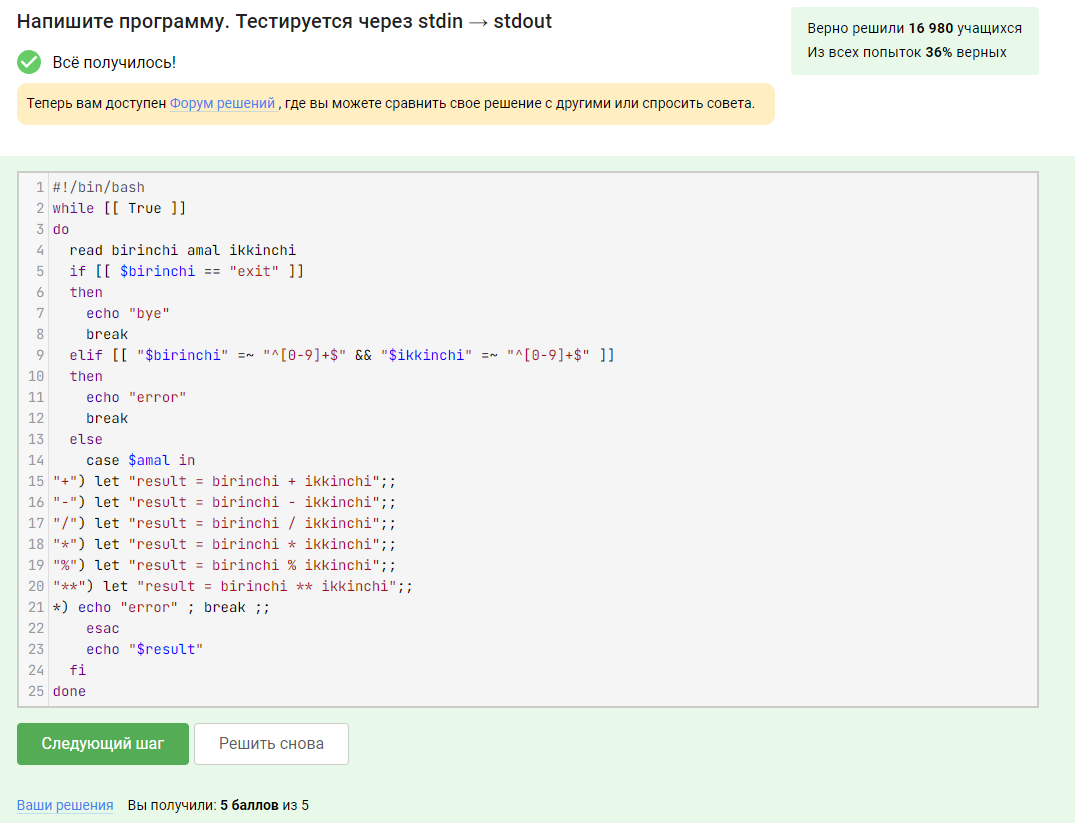


Рис. 21: Задание 18

-iname ищет без учета регистра, а -name в точности как в запросе. Звездочка стоит после слова - это значит после слова может быть сколько угодно символов.

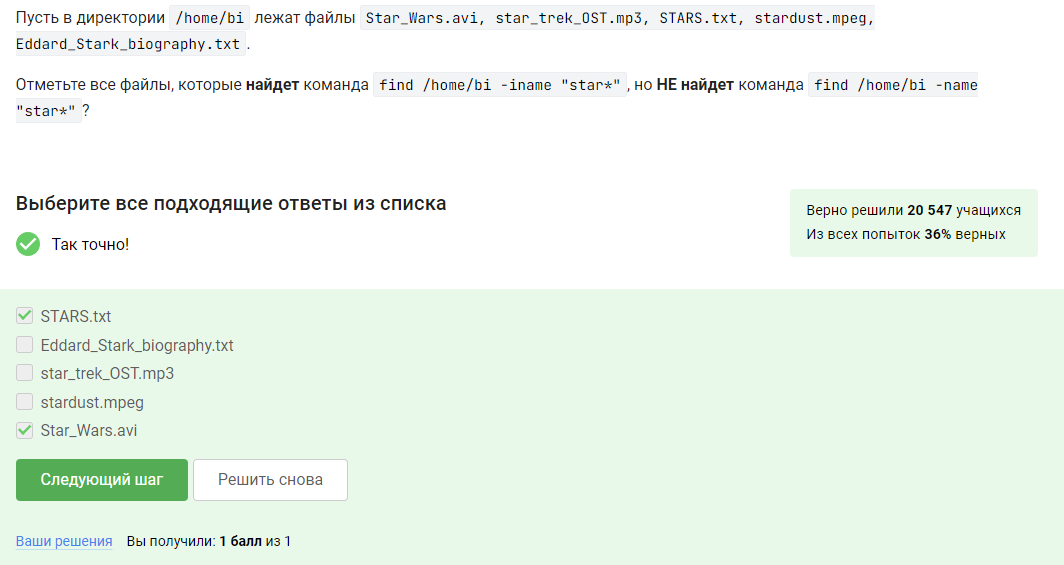


Рис. 22: Задание 19

Проверим каждое утверждение на практике и найдём нужные ответы.

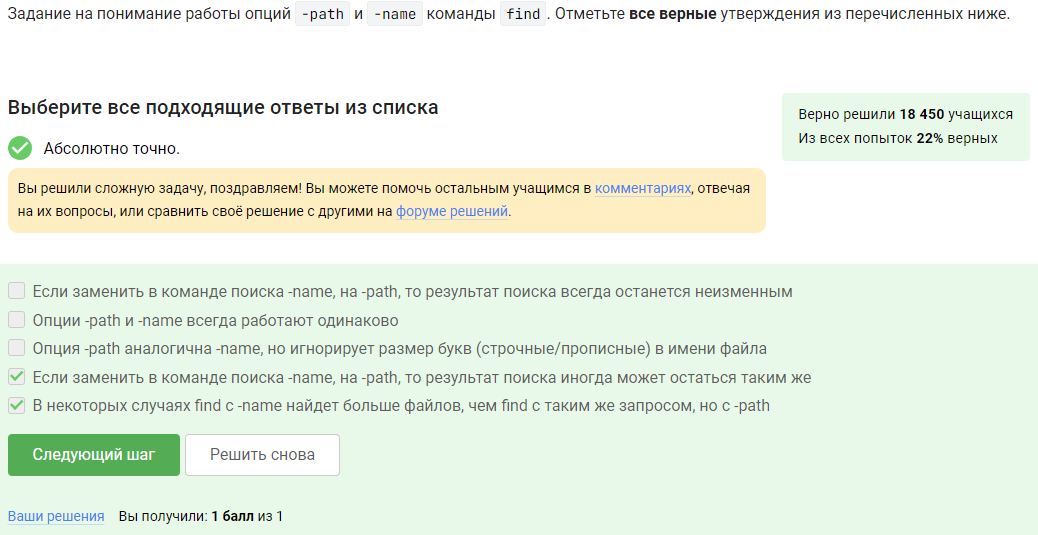


Рис. 23: Задание 20

Текущий каталог - это depth=1  
/home/bi -> depth=1  
/home/bi/dir1 -> depth=2  
/home/bi/dir1/dir2 -> depth=3

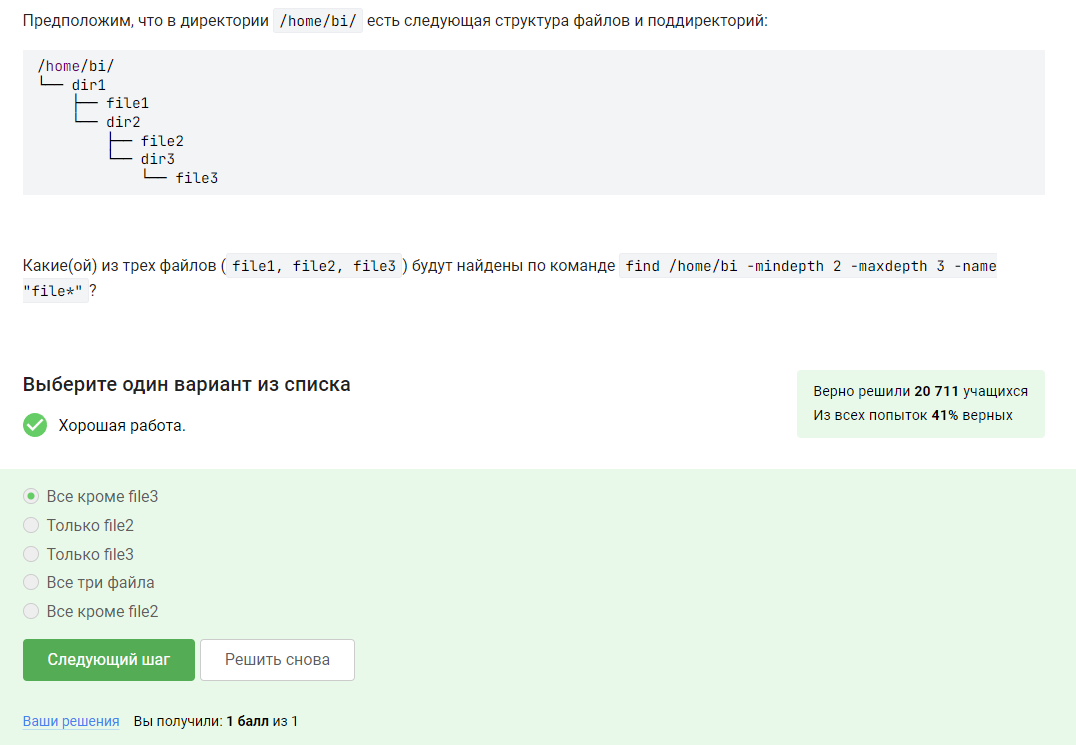


Рис. 24: Задание 21

Из описания man: Print NUM lines of trailing context after/before matching lines “matching lines” - множественное число, строки в которых нашлось совпадение Т.е. если идут 2…10…100 строк подряд, в которых обнаружилось совпадение, контекст будет выведен до и после этой ГРУППЫ строк, а не до и после каждой строки в этой группе.

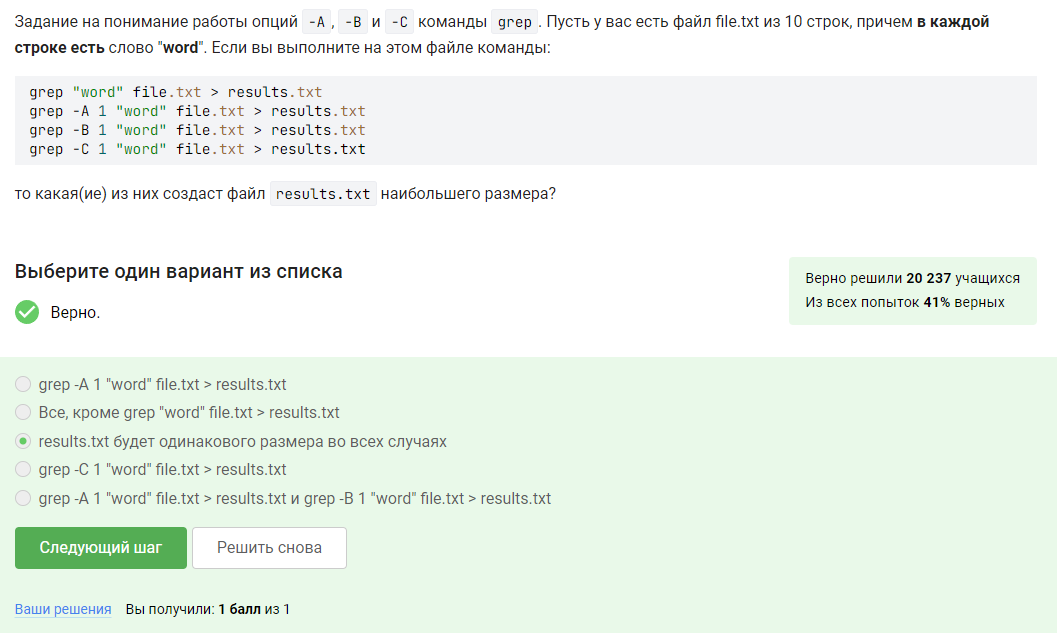


Рис. 25: Задание 22

Снова проверяем на практике.

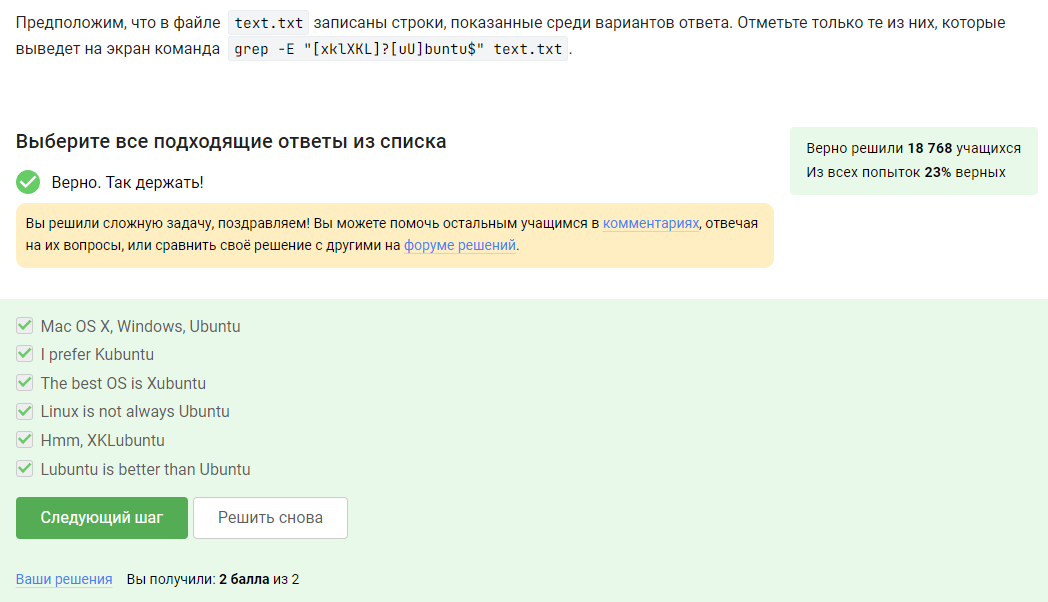


Рис. 26: Задание 23

Напишем без -n и посмотрим.

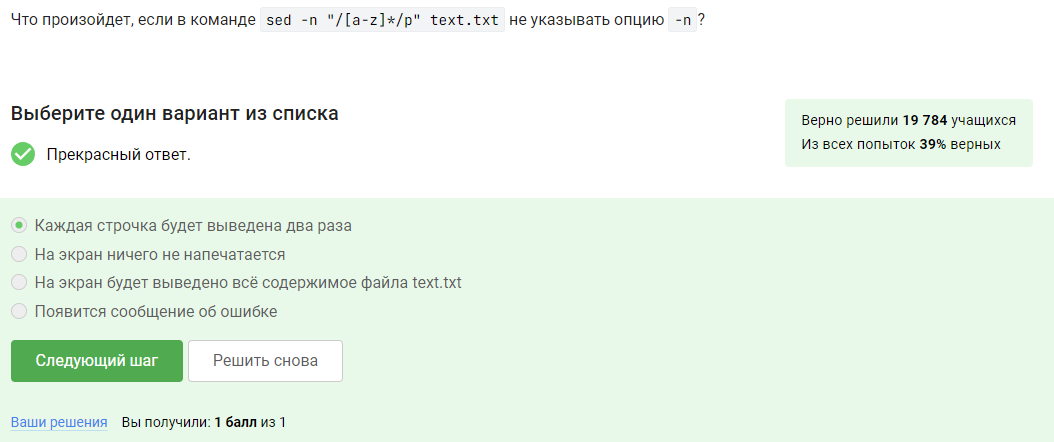


Рис. 27: Задание 24

аббревиатура ABBA отличается от двух других аббревиатур тем, что справа он неё стоит запятая без пробела: “ABBA,”. При этом по условию аббревиатура должна выглядеть как [ XX ] или [ XXX ] (и ещё больше X). Следовательно, для этой проверки надо добавить пробел квадратными скобками [ ] слева и, соответственно, с права.

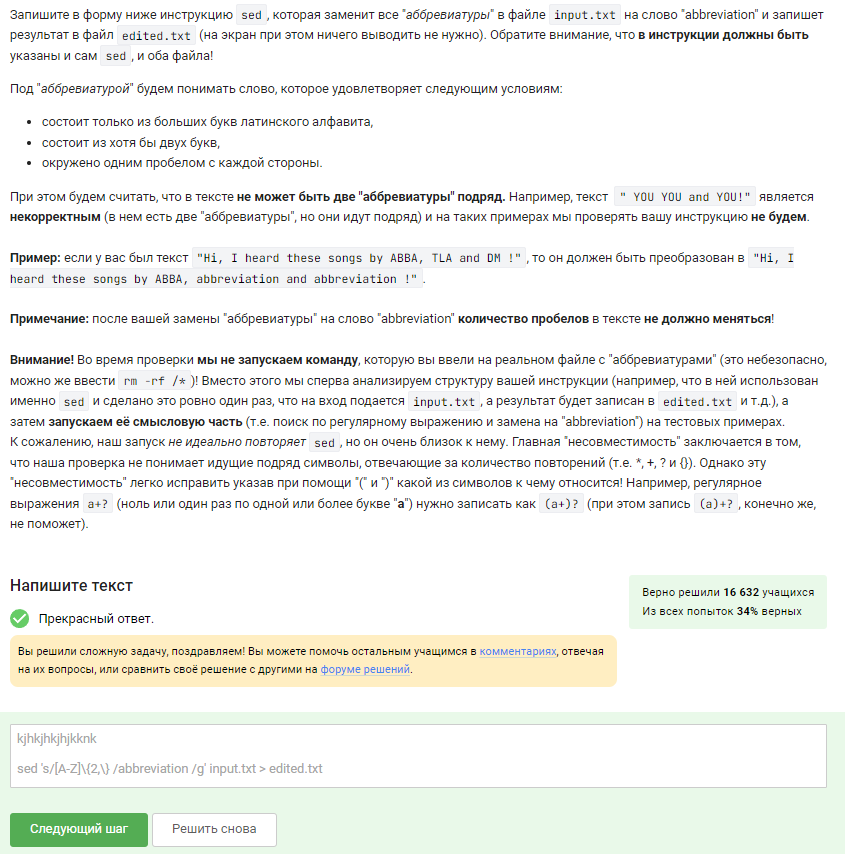


Рис. 28: Задание 25

Перебором посмотрим и отметим нужный вариант.

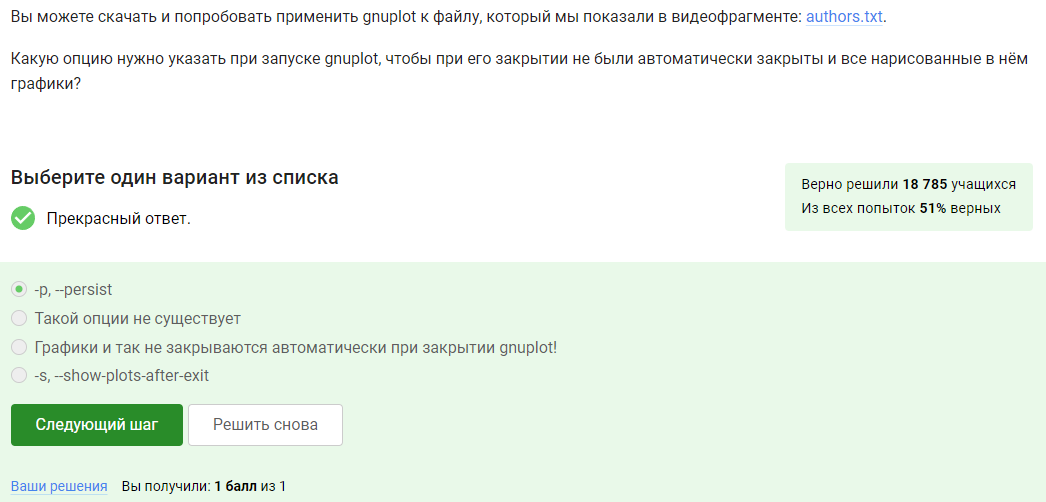


Рис. 29: Задание 26

Формат CSV содержит строки, где столбцы разделены запятой.

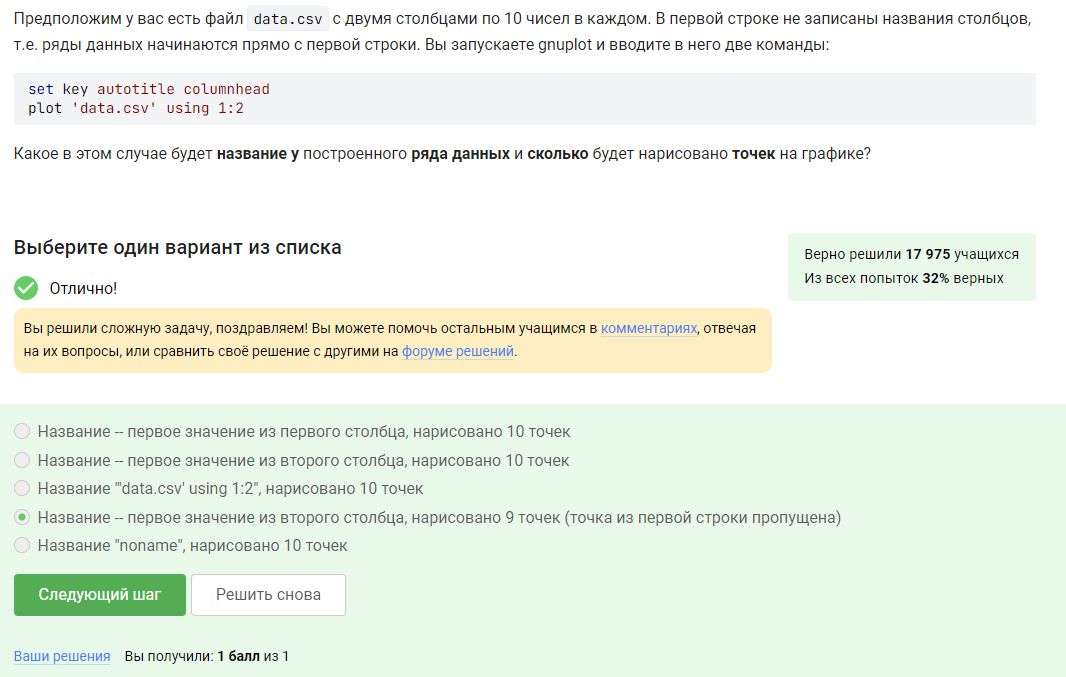


Рис. 30: Задание 27

Cначала идет команда установки подписей, а потом в скобках: подпись - пробел - переменная с координатой - запятая. Повторяется это количество раз соответствующее числу переменных, и без запятой (в случае с последней переменной). А подпись в свою очередь получается конкатенацией текста из задания и переменной с координатой.

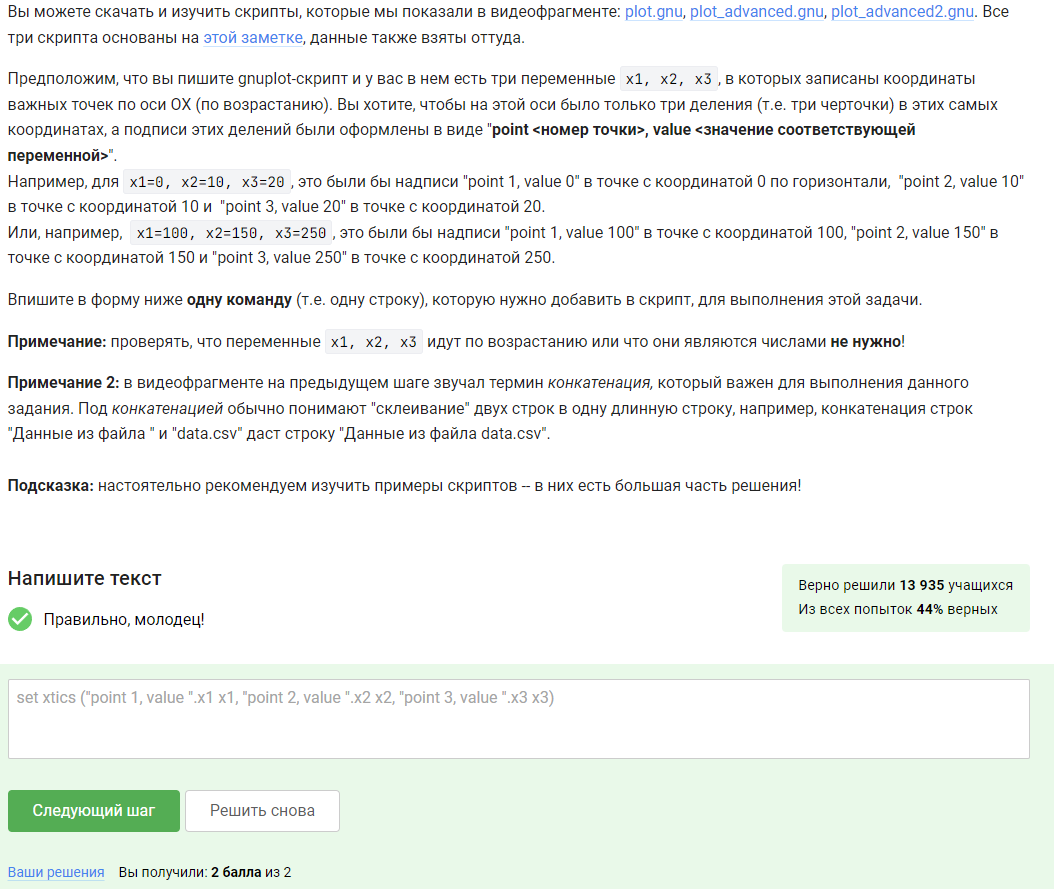


Рис. 31: Задание 28

1. График строится строкой “splot x2+y2”.
2. Вращение задается строкой “zrot=(zrot+10)%360”. Значит, смещение вперед (которое было изначально) можно также задать строкой “zrot=(zrot+360+10)%360” или иначе говоря “zrot=(zrot+370)%360”. А теперь посмотрим на наше требование - чтоб вращалось в другую сторону, значит, по аналогии, необходимо вместо перебора на 10 сделать недобор. “zrot=(zrot+350)%360”.
3. Строка “pause 0.2” ставит выполнение на паузу на определенный промежуток времени. В задании сказали перерисовывать чаще, значит пауза должна быть меньше.

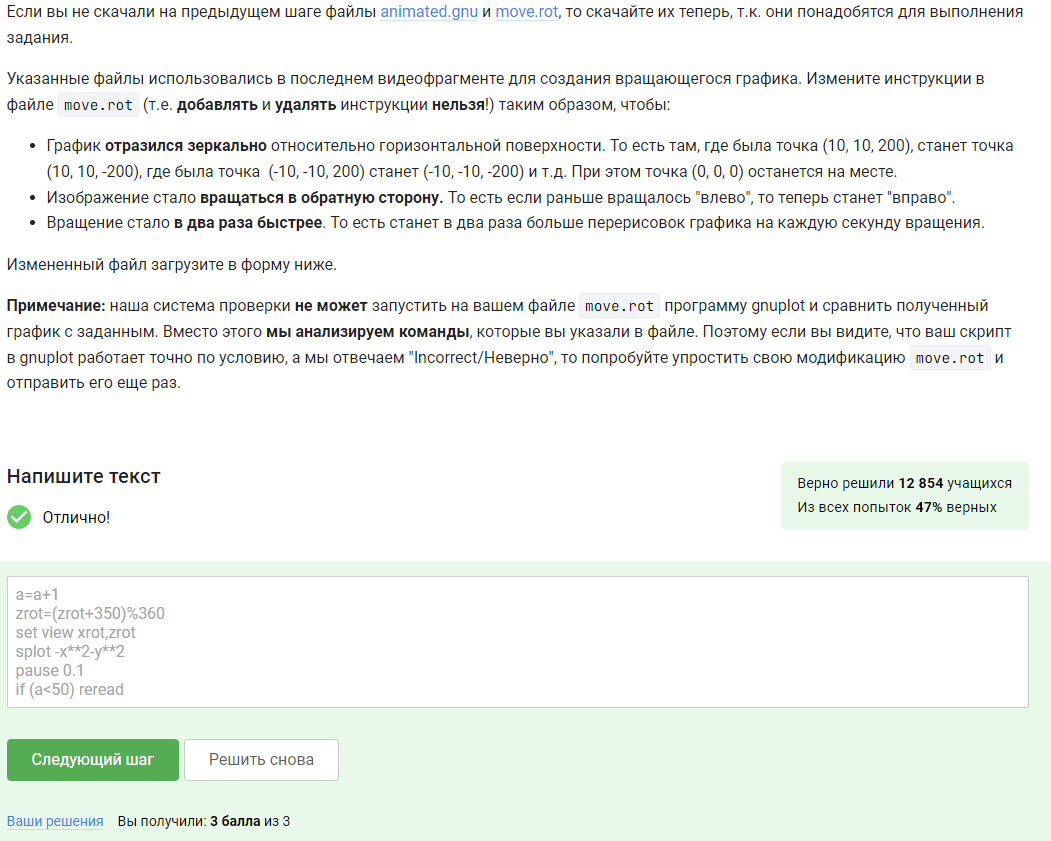


Рис. 32: Задание 29

• r - чтение; • w - запись; • x - выполнение; • s - выполнение от имени суперпользователя; • u - владелец файла; • g - группа файла; • o - все остальные пользователи; • 0 - никаких прав;

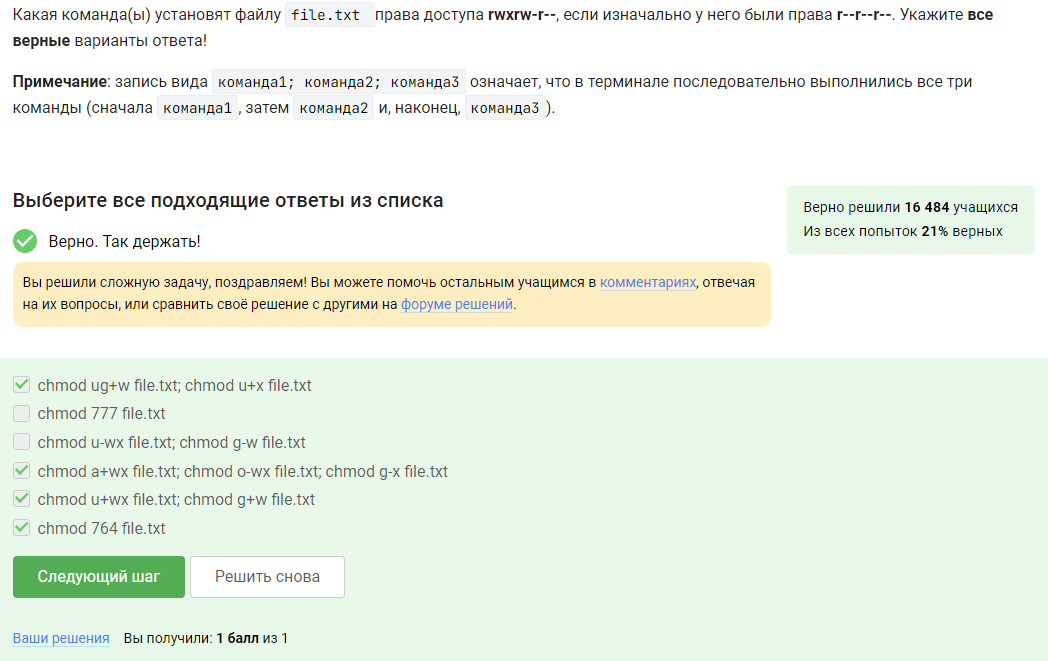


Рис. 33: Задание 30

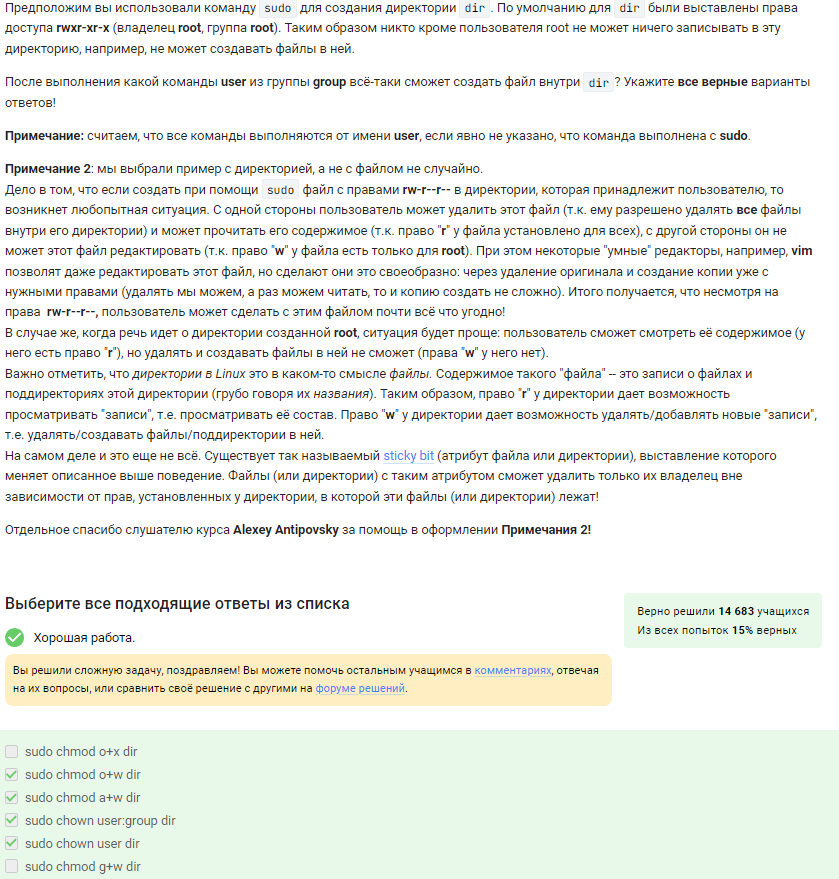


Рис. 34: Задание 31

• wc -l вывести количество строк • wc -c вывести количество байт • wc -m вывести количество символов • wc -L вывести длину самой длинной строки • wc -w вывести количество слов

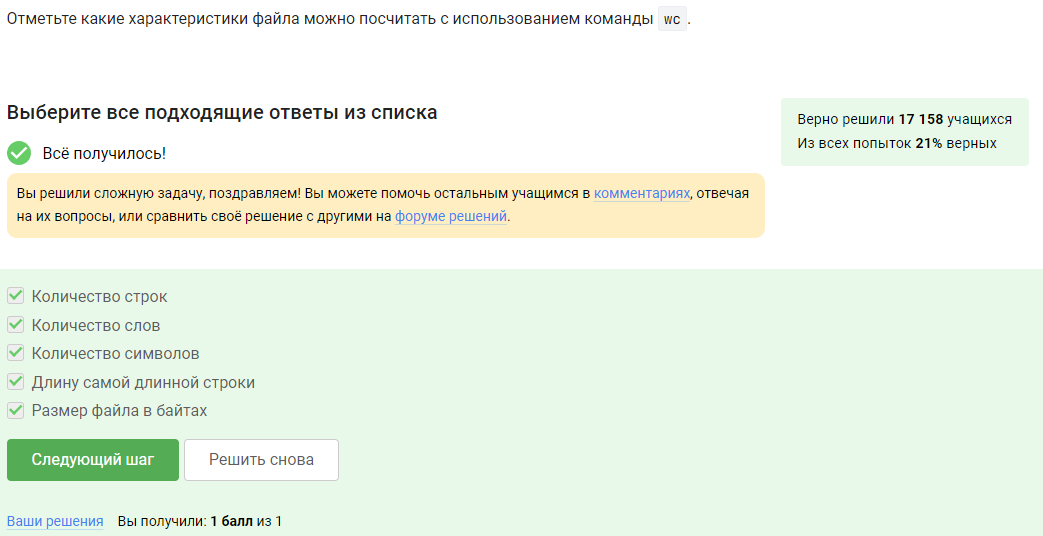


Рис. 35: Задание 32

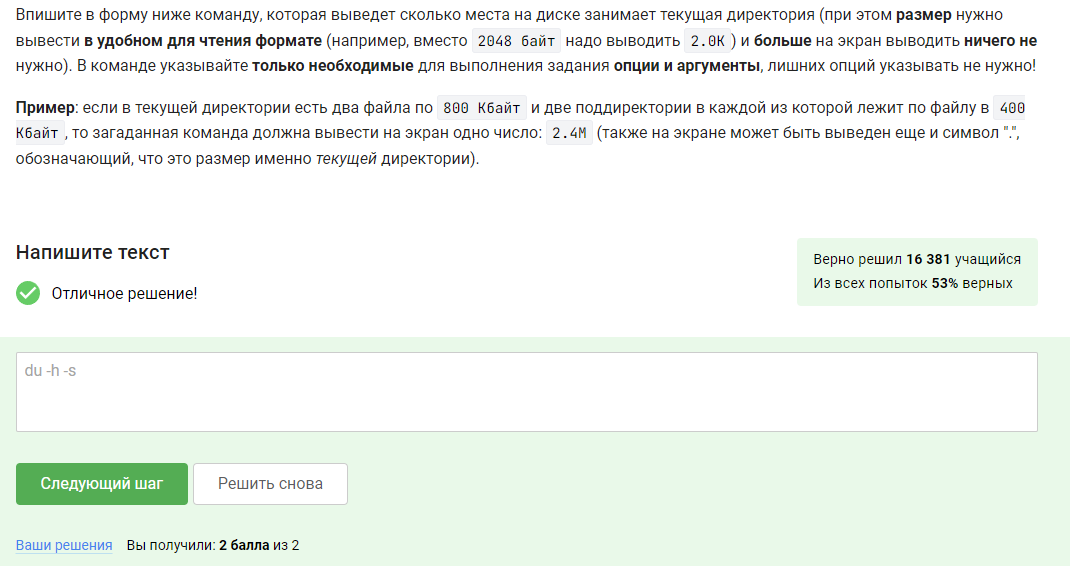


Рис. 36: Задание 33

Создаётся три директории: dir1, dir2 и dir3.

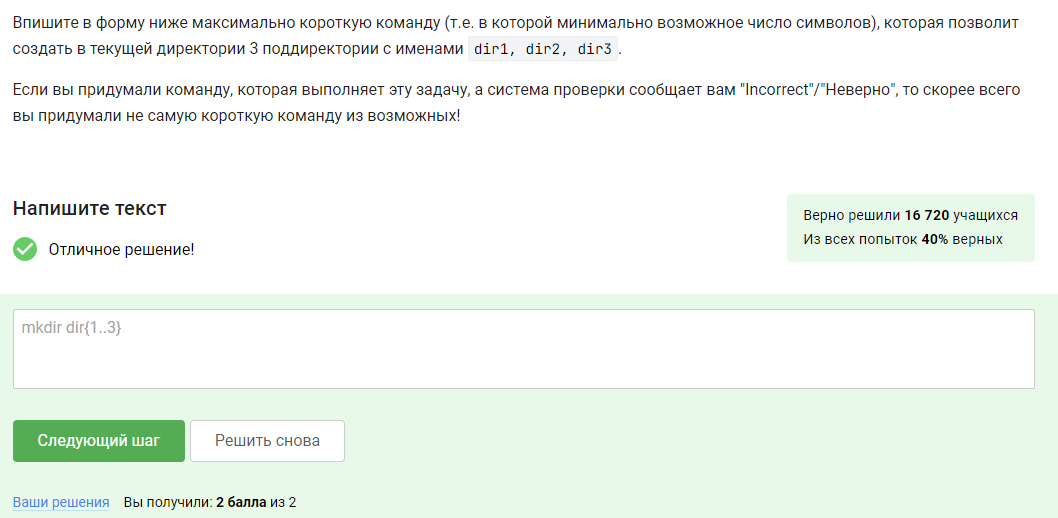


Рис. 37: Задание 34

# 4 Выводы

В ходе выполнения работы мы просмотрели курс и узнали/вспомнили возможности операционной системы Linux

# Список литературы

::: Stepik