Выполнение внешнего курса. Блок №3. Продвинутые темы

Отчёт

Сергеев Даниил Олегович

Содержание

# 1 Цель работы

Получить основные навыки работы с ОС Linux путем прохождения внешнего курса на образовательной платформе stepik [1].

# 2 Задание

* Пройти курс.
* Получить сертификат.
* Записать видео по каждому разделу.
* Записать итоговую презентацию по каждому этапу.
* Написать отчёт по прохождению контрольных мероприятий по каждому разделу.

# 3 Выполнение лабораторной работы

Приступим к выполнению третьего блока заданий внешнего курса – Продвинутым темам.

## 3.1 Текстовый редактор vim

1. Вопрос 1-й: Из только что открытого в vim файла можно выйти перейдя в командный режим с помощью : и написав q, затем ENTER. (если не производилось изменение файла)

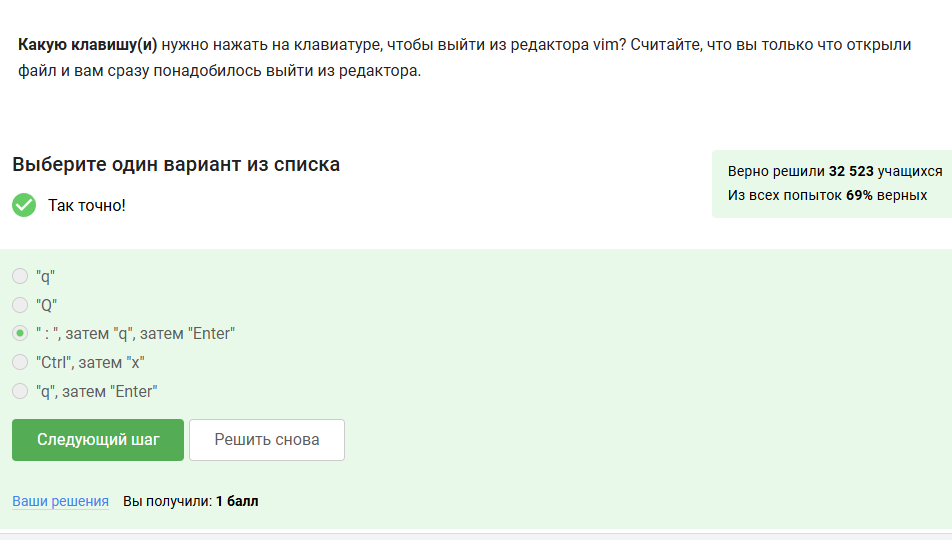


Рис. 1: Задание №1. Условие и ответ

1. Вопрос 2-й: Введем тестовую строку и проведем на ней указанные операции, выберем правильные ответы.

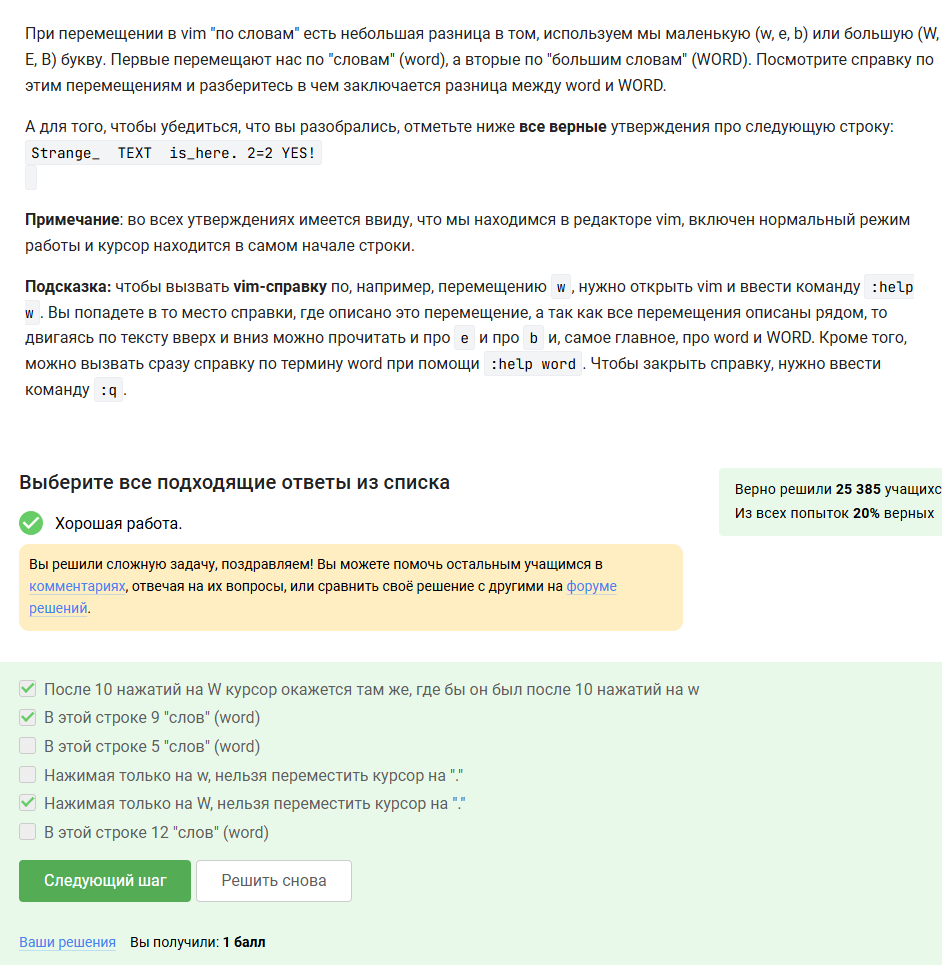


Рис. 2: Задание №2. Условие и ответ

1. Вопрос 3-й: Проверим каждую комбинацию в редакторе vim. Укажем правильные ответы.

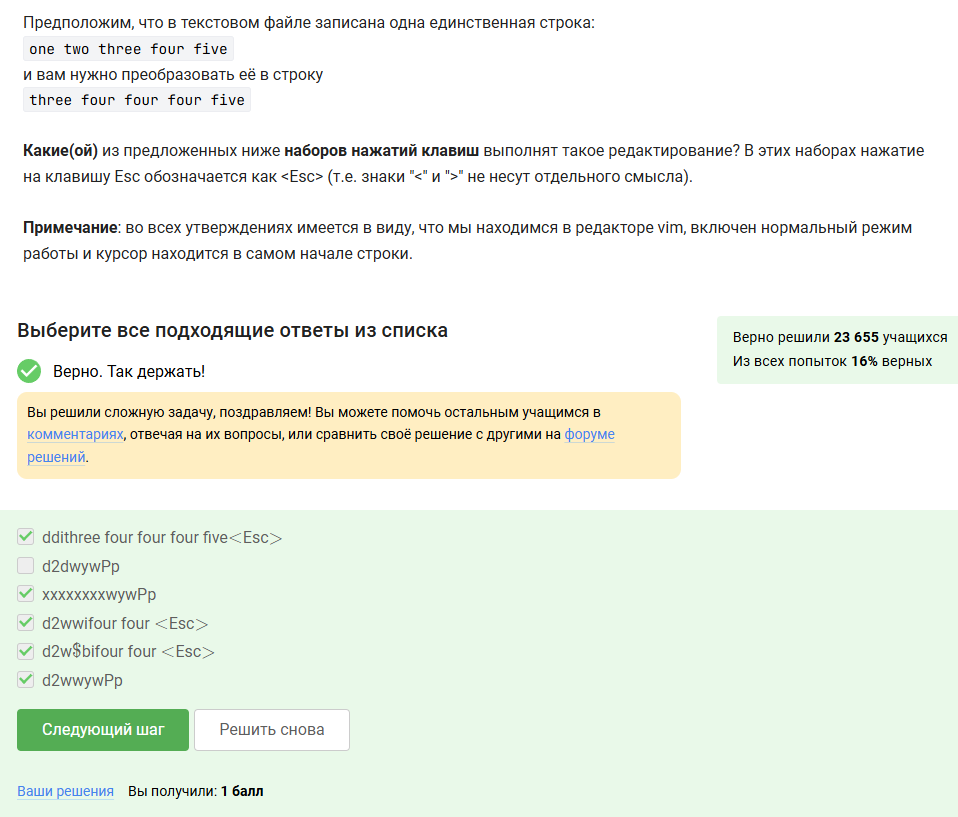


Рис. 3: Задание №3. Условие и ответ

1. Вопрос 4-й: В командном режиме можно совершить замену первого вхождения слова в строке. Для этого укажем строку :%s/Windows/Linux, где:

* % – диапазон (всего файла)
* s – команда замены
* Windows – то что заменяем
* Linux – то на что заменяем

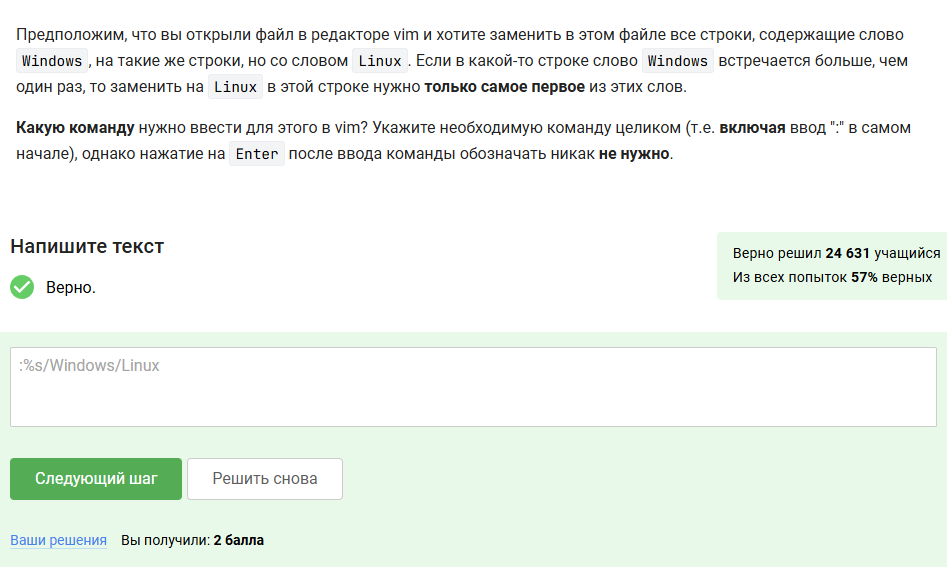


Рис. 4: Задание №4. Условие и ответ

1. Вопрос 5-й: Проверим каждую опцию из предложенного списка, выберем правильные ответы.

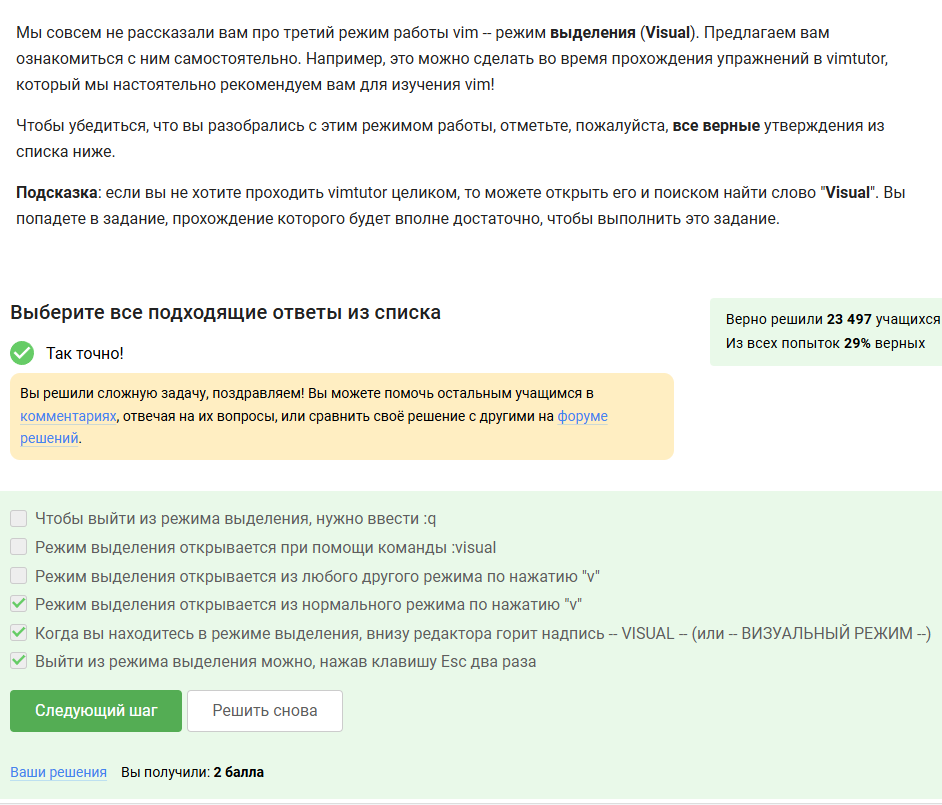


Рис. 5: Задание №5. Условие и ответ

## 3.2 Скрипты на bash: основы

1. Вопрос 6-й: Каждая оболочка хранит свойфайл независимо, поэтому в последней оболочке будут лишь команды C1, C2, C3.

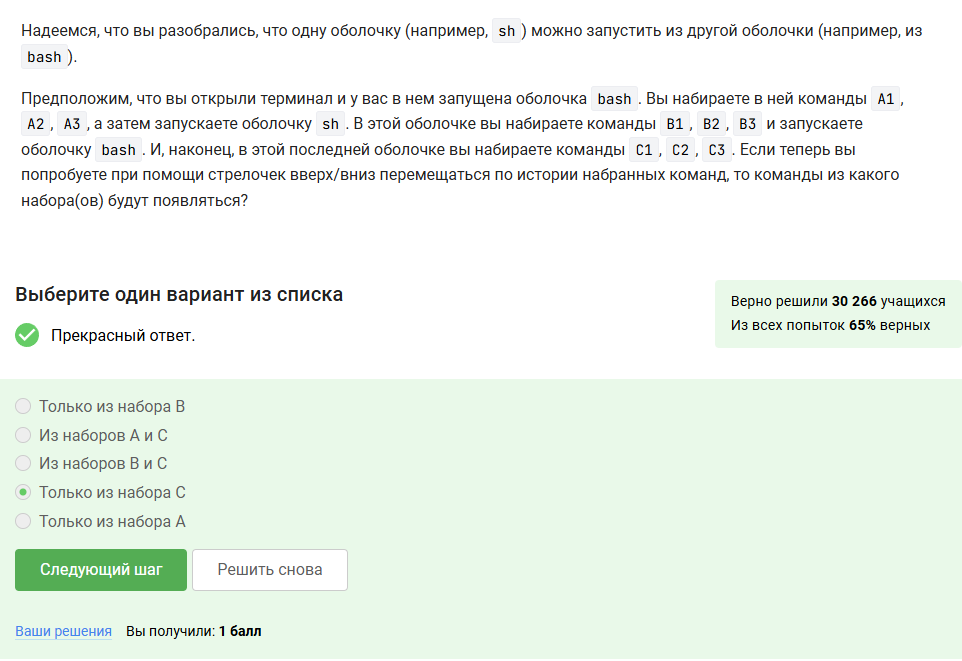


Рис. 6: Задание №6. Условие и ответ

1. Вопрос 7-й: Исходя из скрипта, мы перейдем в домашнюю директорию, после в ней создастся файл file1.txt и команда отправит нас в каталог /home/bi/Desktop. В итоге выбираем второй вариант.

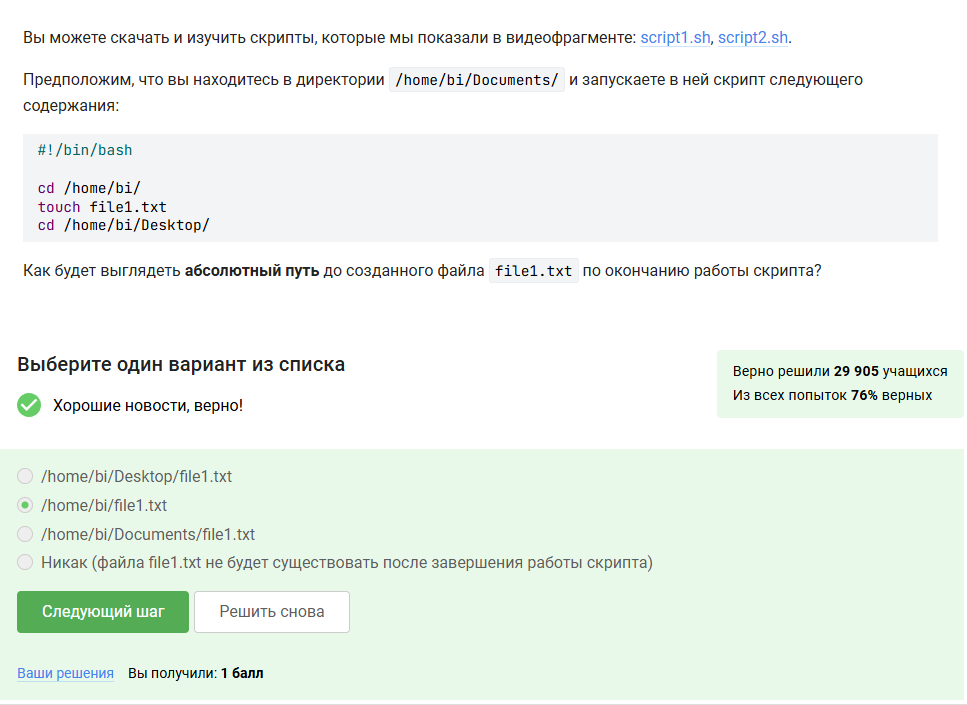


Рис. 7: Задание №7. Условие и ответ

1. Вопрос 8-й: В языке программирования bash в названия переменных допускаются нижние подчёркивания, числа и буквы, однако имена не должны начинаться с цифры. Выберем подходящие варианты ответа.

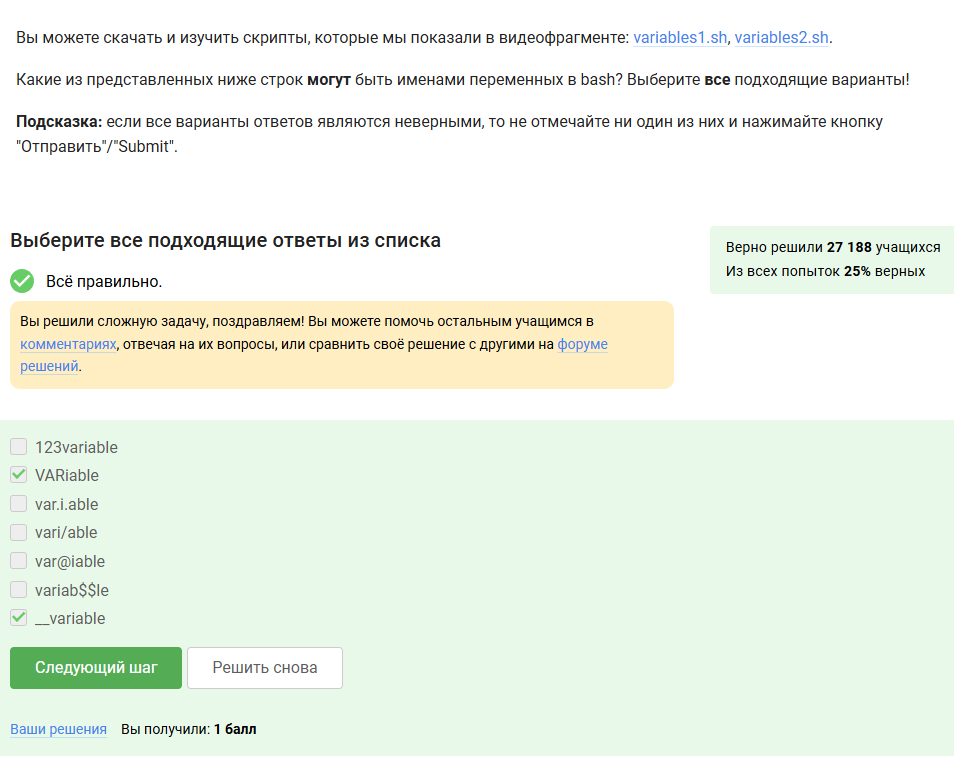


Рис. 8: Задание №8. Условие и ответ

1. Вопрос 9-й: Используем команду echo для вывода сообщения. Чтобы символ доллара корректно отображался, укажем обратный слеш перед ним: ...\$1... .

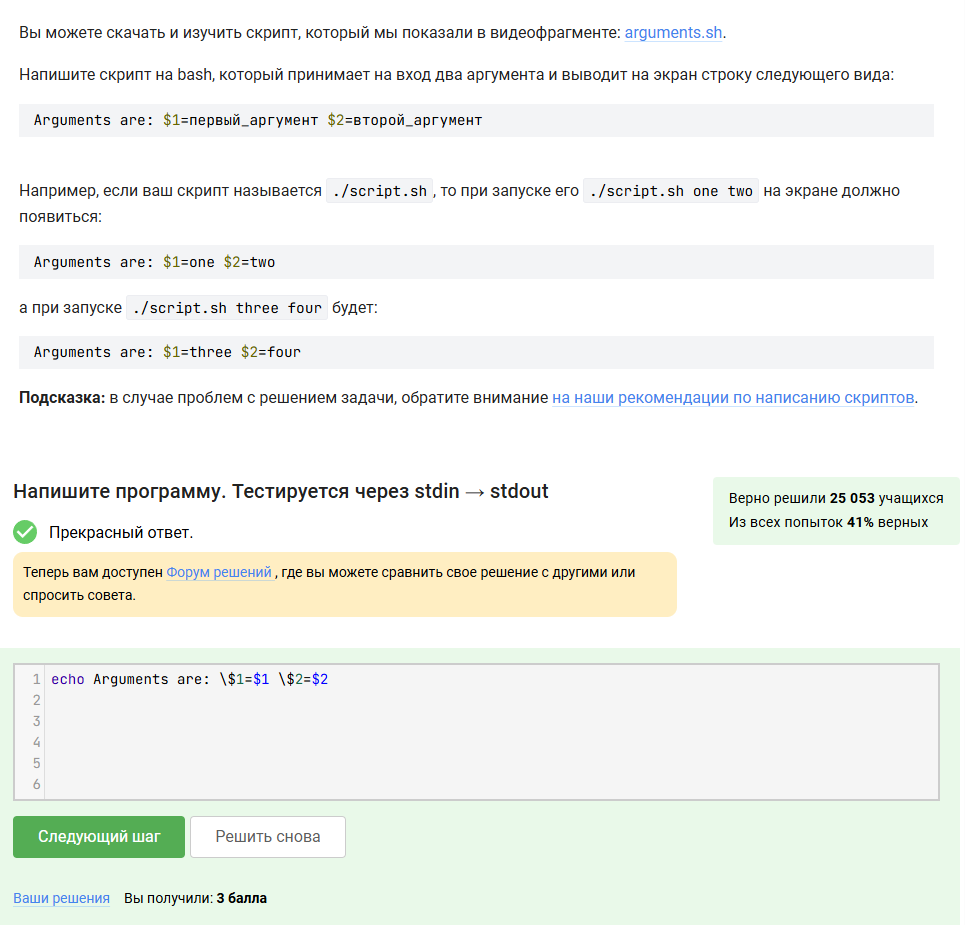


Рис. 9: Задание №9. Условие и ответ

**Листинг №3.2.1. Команда вывода сообщения.**

echo Arguments are: \$1=$1 \$2=$2

## 3.3 Скрипты на bash: ветвления и циклы

1. Вопрос 10-й: Двойные квадратные скобки – усовершенствованная версия тестовой конструкции test, позволяющая совершать сравнения и тесты. Изучим возможные опции для этой конструкции и выберем подходящие варианты.

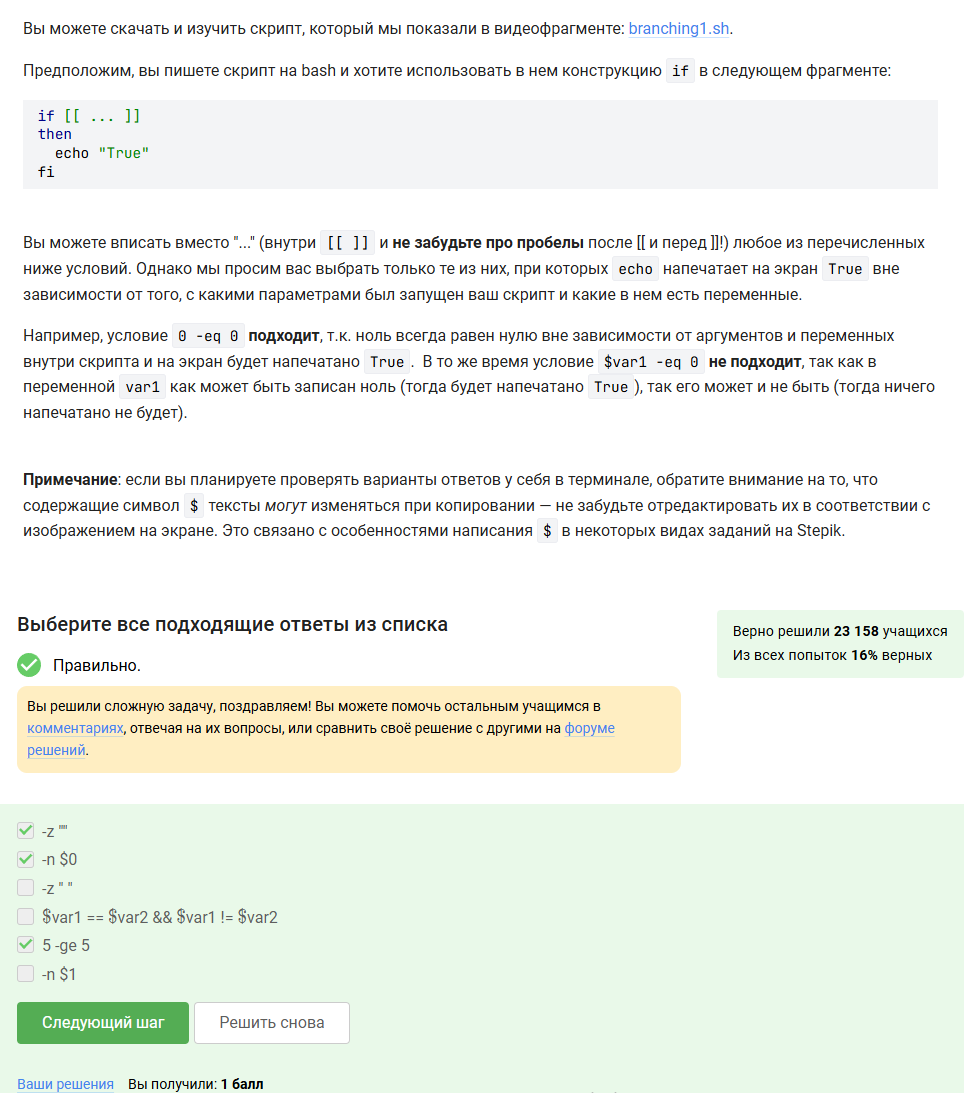


Рис. 10: Задание №10. Условие и ответ

1. Вопрос 11-й: Задание проверяет знание опций для [[]]. Пройдемся по условиям с заданными параметрами и выберем правильные варианты из списка.

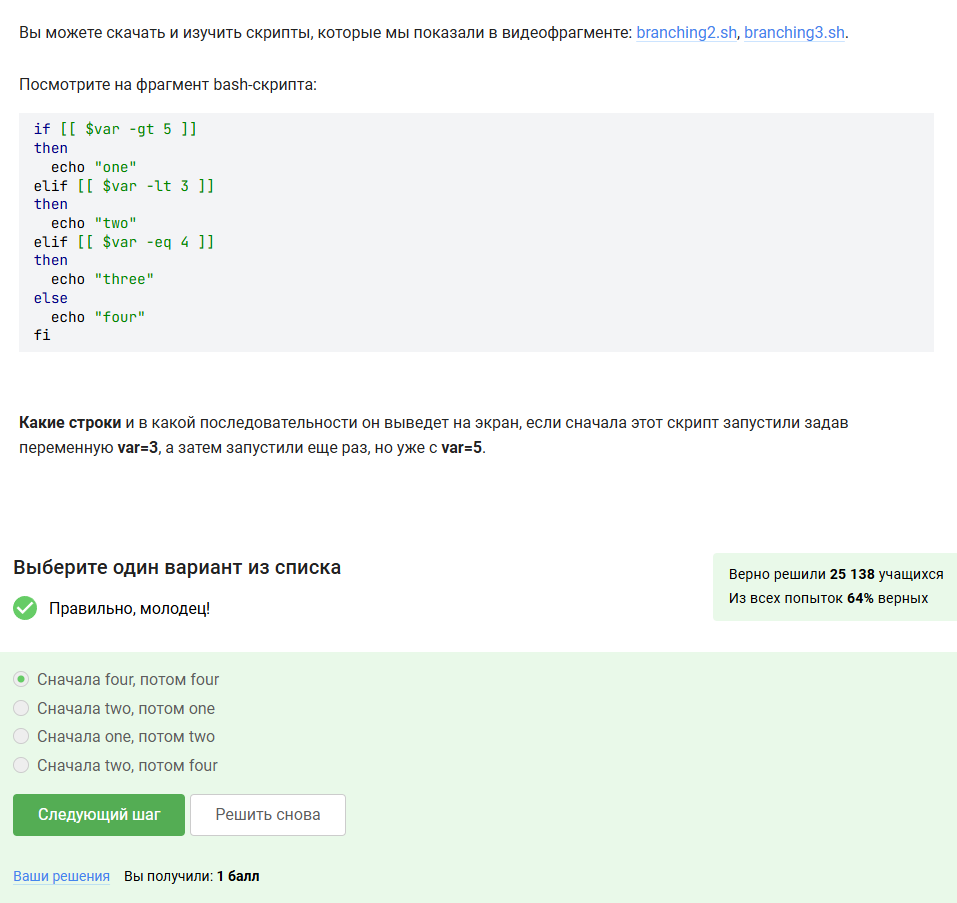


Рис. 11: Задание №11. Условие и ответ

1. Вопрос 12-й: Будем считывать первый аргумент с помощью $1. Для проходки по возможным вариантам используем оператор case.

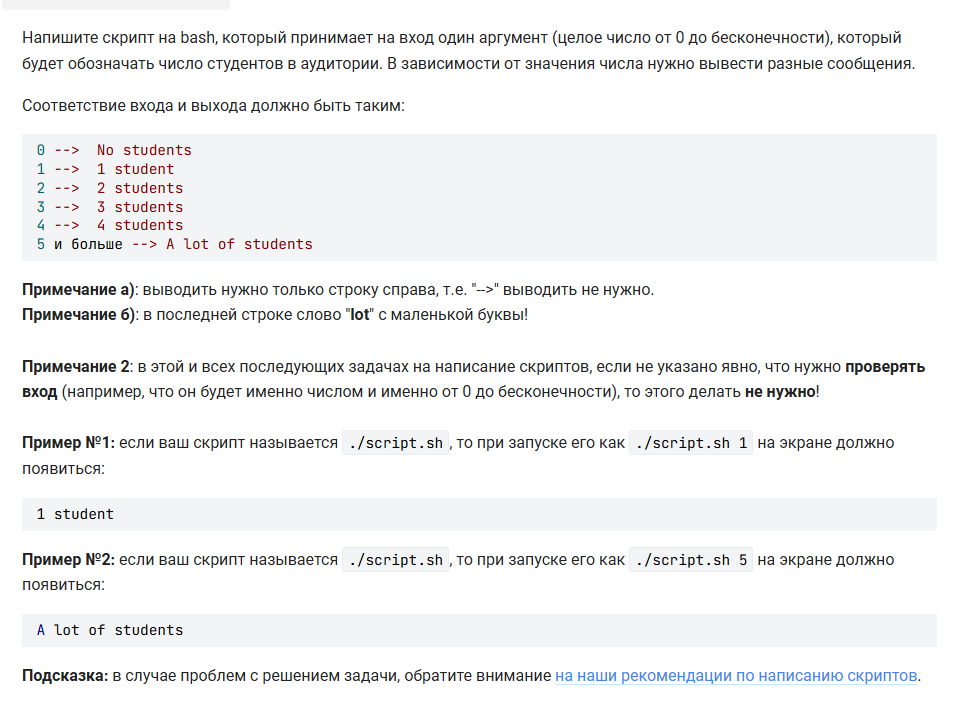


Рис. 12: Задание №12. Условие

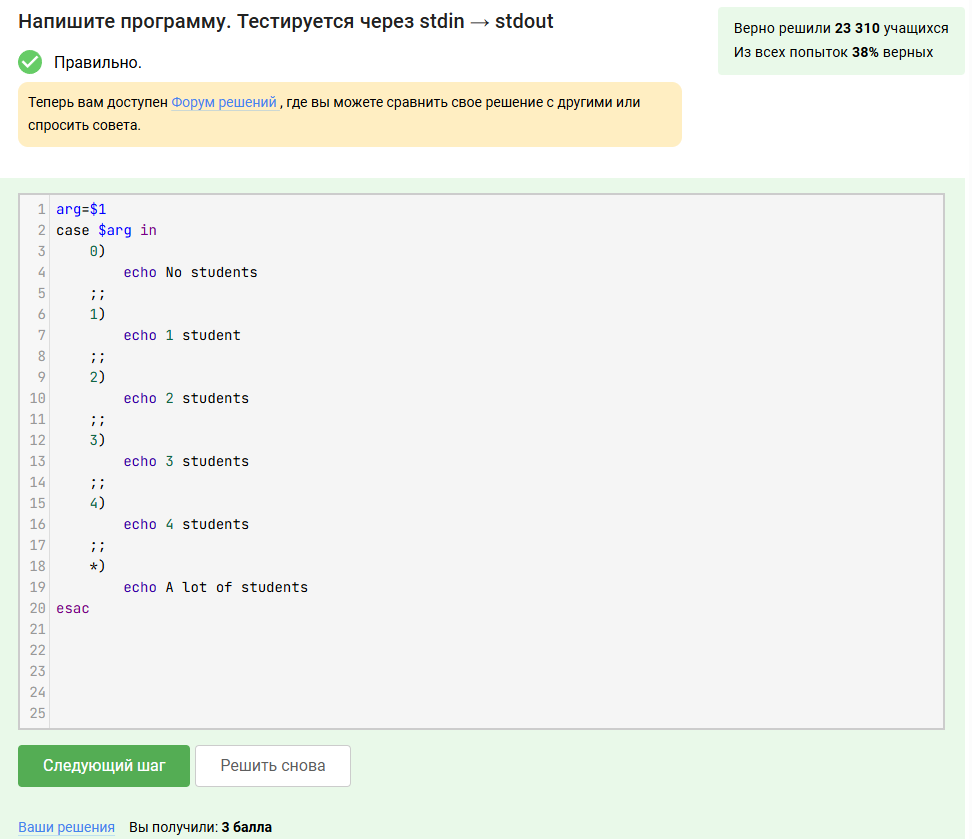


Рис. 13: Задание №12. Ответ

**Листинг №3.3.1. Скрипт проверки количества студентов.**

arg=$1  
case $arg in  
 0)  
 echo No students  
 ;;  
 1)  
 echo 1 student   
 ;;  
 2)  
 echo 2 students  
 ;;  
 3)  
 echo 3 students  
 ;;  
 4)  
 echo 4 students  
 ;;  
 \*)  
 echo A lot of students  
esac

1. Вопрос 13-й: Команда запуститься 5 раз, так как знак запятой тоже считается отдельным значением переменной str. Условие [[ $str > "c" ]] проверяет по длине строк, поэтому команда закончится на последнем аргументе, не выводя сообщение finish.

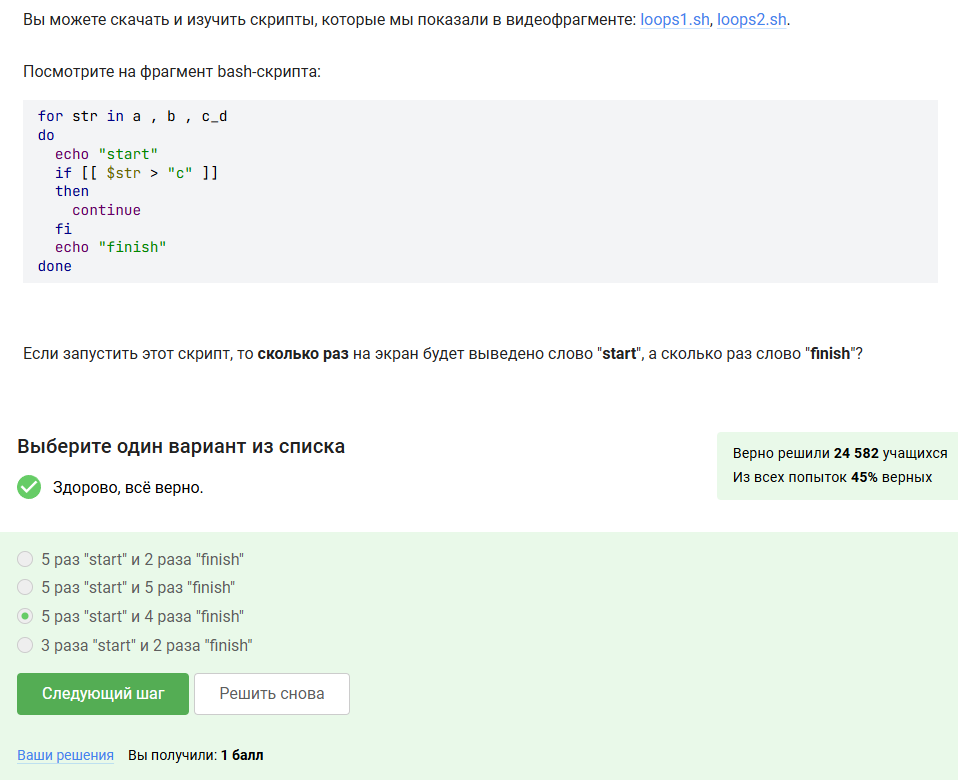


Рис. 14: Задание №13. Условие и ответ

1. Вопрос 14-й: Программу запишем в цикл while true. Сделаем отдельные переменные для имени и возраста, считаем значения через команду read, проверяя на пустую строку либо нулевой возраст.



Рис. 15: Задание №14. Условие

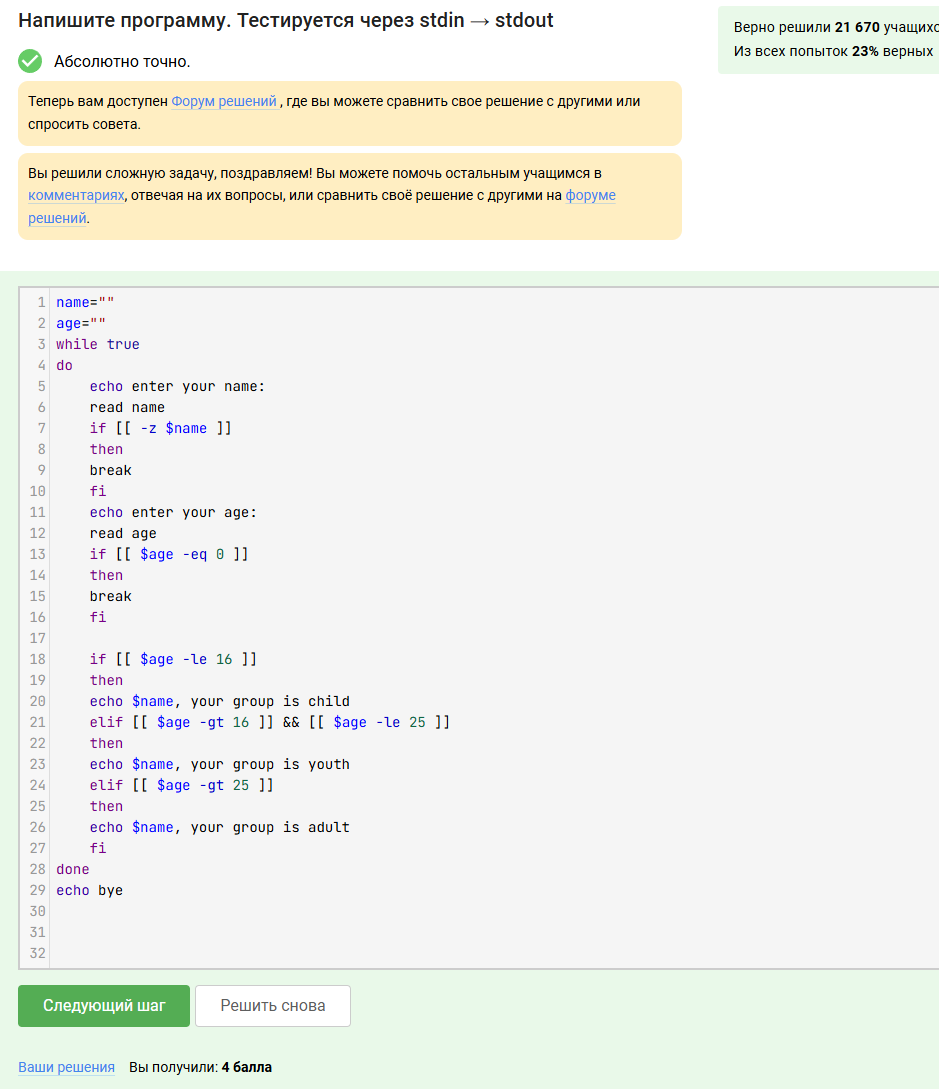


Рис. 16: Задание №14. Ответ

**Листинг №3.3.2. Скрипт проверки возраста.**

name=""  
age=""  
while true  
do  
 echo enter your name:  
 read name  
 if [[ -z $name ]]  
 then  
 break  
 fi  
 echo enter your age:  
 read age  
 if [[ $age -eq 0 ]]  
 then  
 break  
 fi  
   
 if [[ $age -le 16 ]]  
 then  
 echo $name, your group is child  
 elif [[ $age -gt 16 ]] && [[ $age -le 25 ]]  
 then  
 echo $name, your group is youth  
 elif [[ $age -gt 25 ]]  
 then  
 echo $name, your group is adult  
 fi  
done  
echo bye

## 3.4 Скрипты на bash: разное

1. Вопрос 15-й: Выражения с командой let можно писать как в кавычках, так и без них. Во втором варианте происходит сложение (объединение) строк, а первый вариант не сработает из-за неправильного операнда.

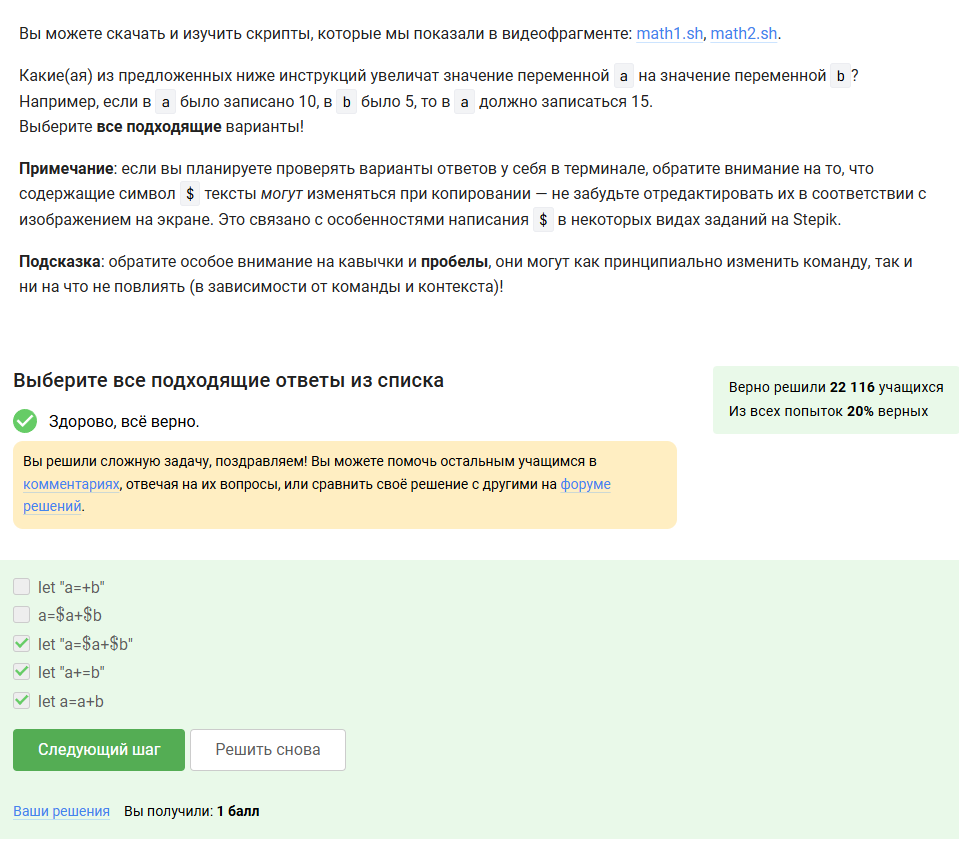


Рис. 17: Задание №15. Условие и ответ

1. Вопрос 16-й: Обратные одинарные ковычки позволяют выполнять команды внутри сценария оболочки, поэтому вывод будет равен пути pwd, указанном в файле.

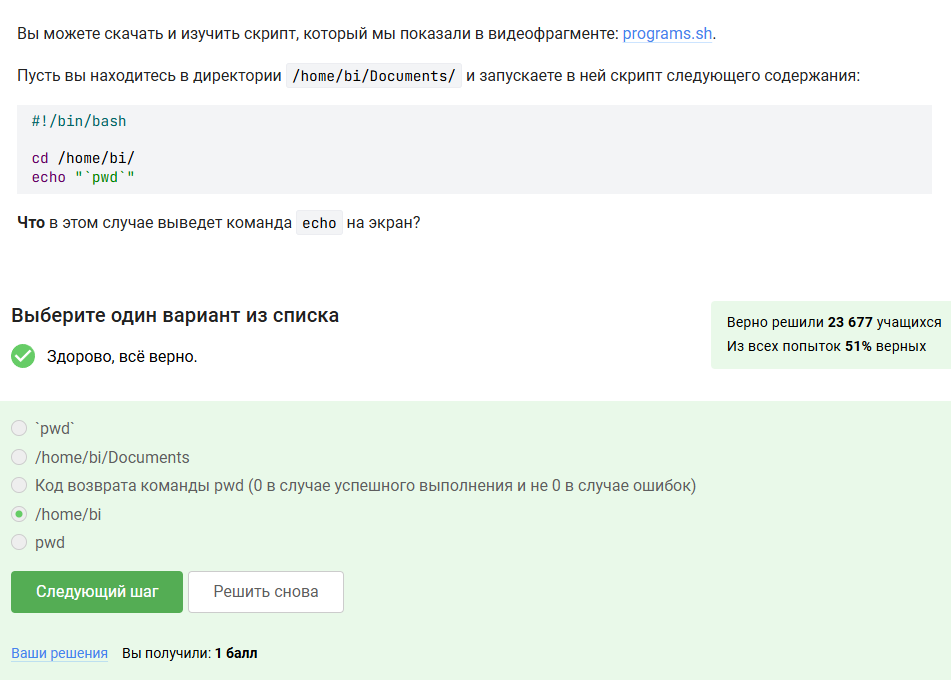


Рис. 18: Задание №16. Условие и ответ

1. Вопрос 17-й: Достать вывод потока ошибок можно через встроенную переменную $? либо через запись результата в какой-нибудь файл с последующим считыванием кода выхода уже этого условия.

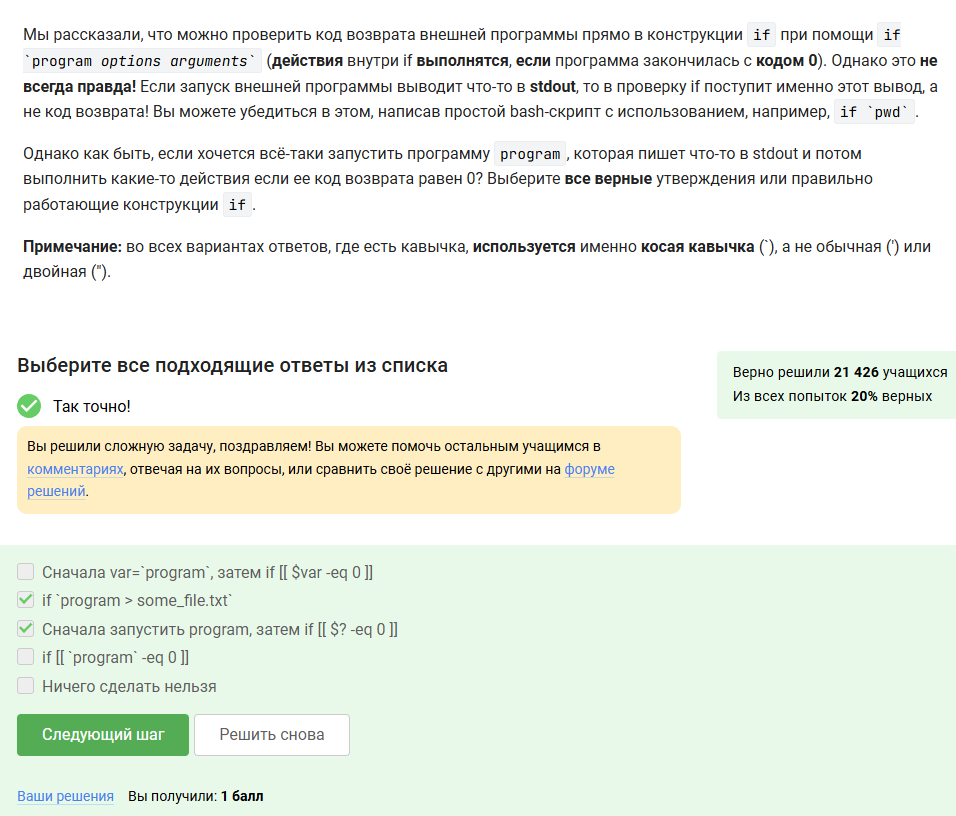


Рис. 19: Задание №17. Условие и ответ

1. Вопрос 18-й: Подвох этой задачи заключается в том, что мы пытаемся вывести значение переменной $c1, хотя она является локальной в теле функции counter. На выходе её значение будет равно пустой строке.

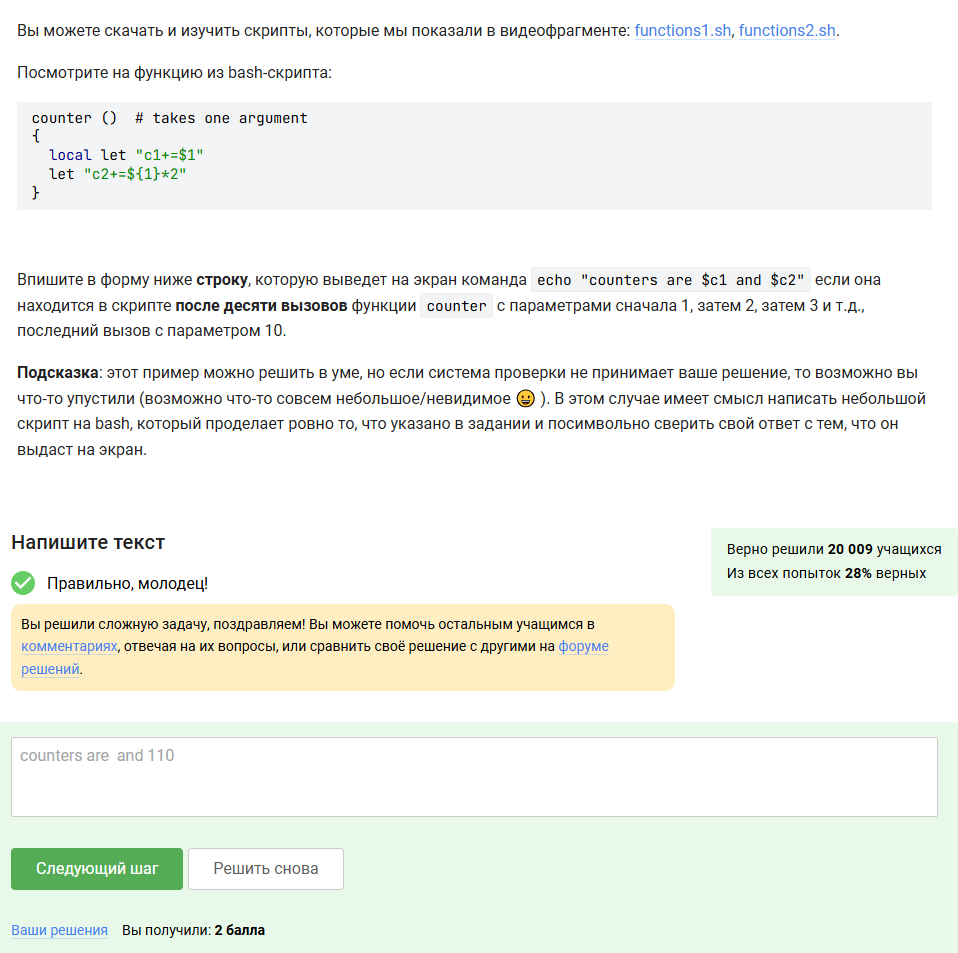


Рис. 20: Задание №18. Условие и ответ

1. Вопрос 19-й: Напишем программу для вычисления НОД, используя алгоритм Евклида и рекурсивный вызов функции в bash.

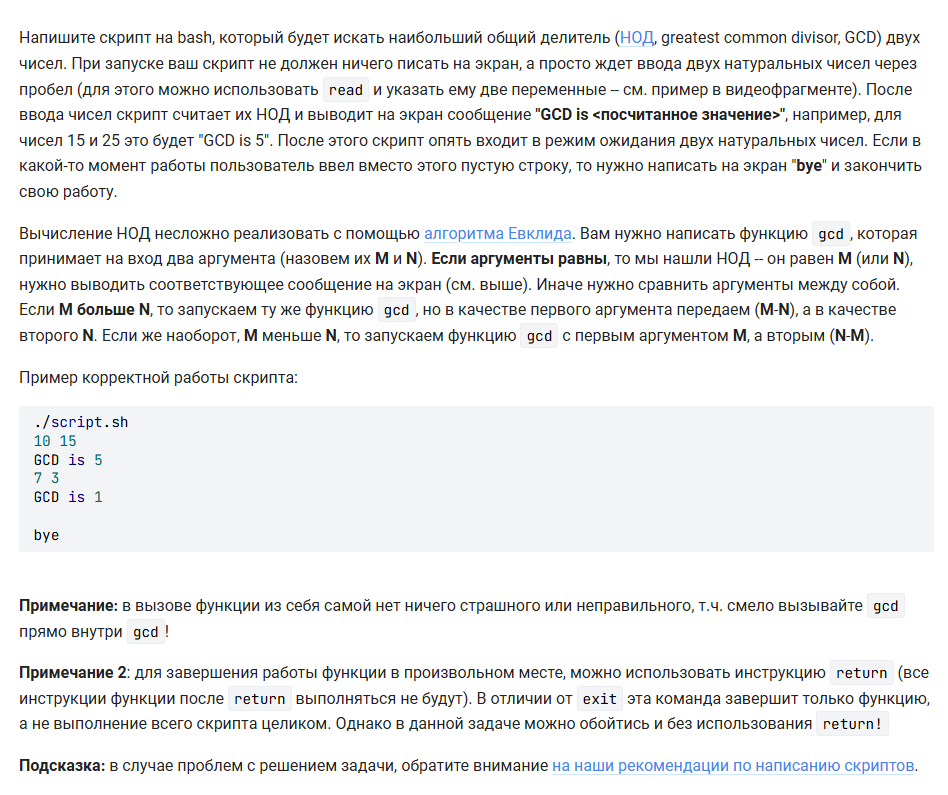


Рис. 21: Задание №19. Условие

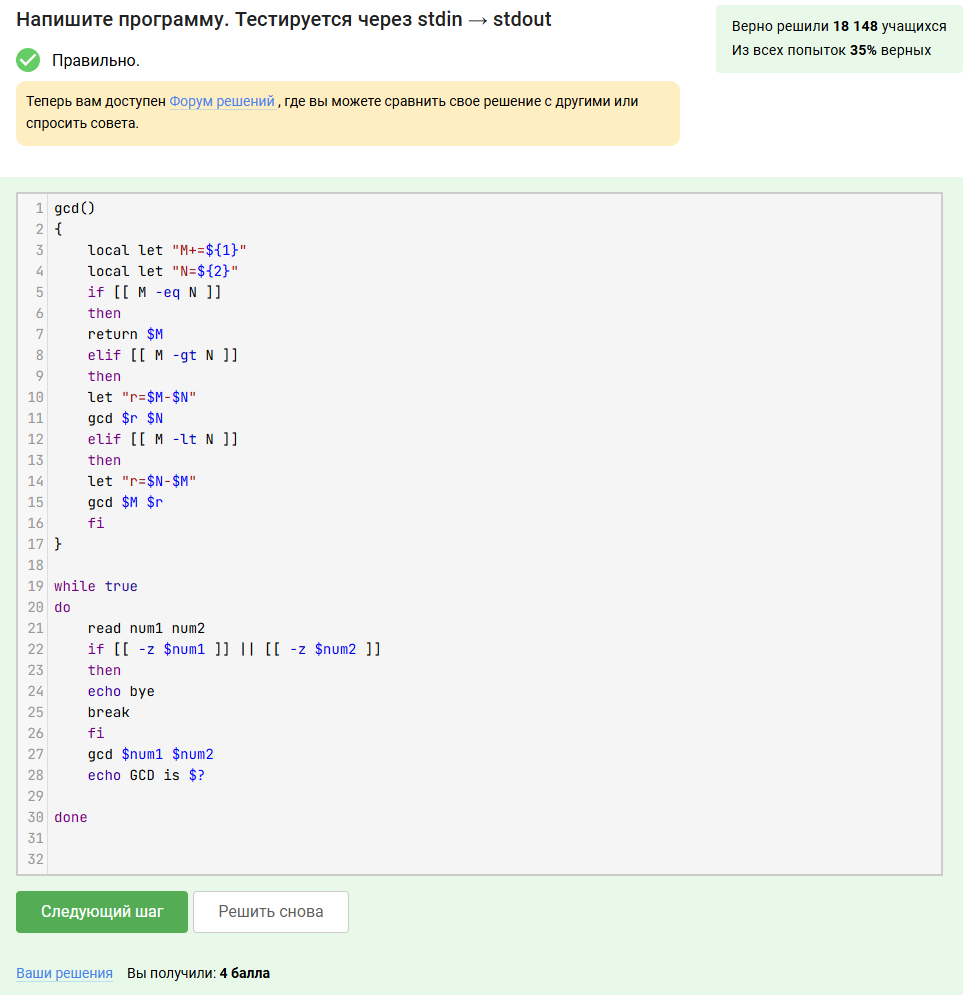


Рис. 22: Задание №19. Ответ

**Листинг №3.4.1. Скрипт нахождения НОД.**

gcd()  
{  
 local let "M+=${1}"  
 local let "N=${2}"  
 if [[ M -eq N ]]  
 then  
 return $M  
 elif [[ M -gt N ]]  
 then  
 let "r=$M-$N"  
 gcd $r $N  
 elif [[ M -lt N ]]  
 then  
 let "r=$N-$M"  
 gcd $M $r  
 fi   
}  
  
while true  
do  
 read num1 num2  
 if [[ -z $num1 ]] || [[ -z $num2 ]]  
 then  
 echo bye  
 break  
 fi  
 gcd $num1 $num2  
 echo GCD is $?  
done

1. Вопрос 20-й: Реализуем калькулятор. Будем считывать уже 3 аргумента командой read. Чтобы оператор case не путал умножение "\*") и оставшиеся варианты \*), поместим знак умножения в кавычки.

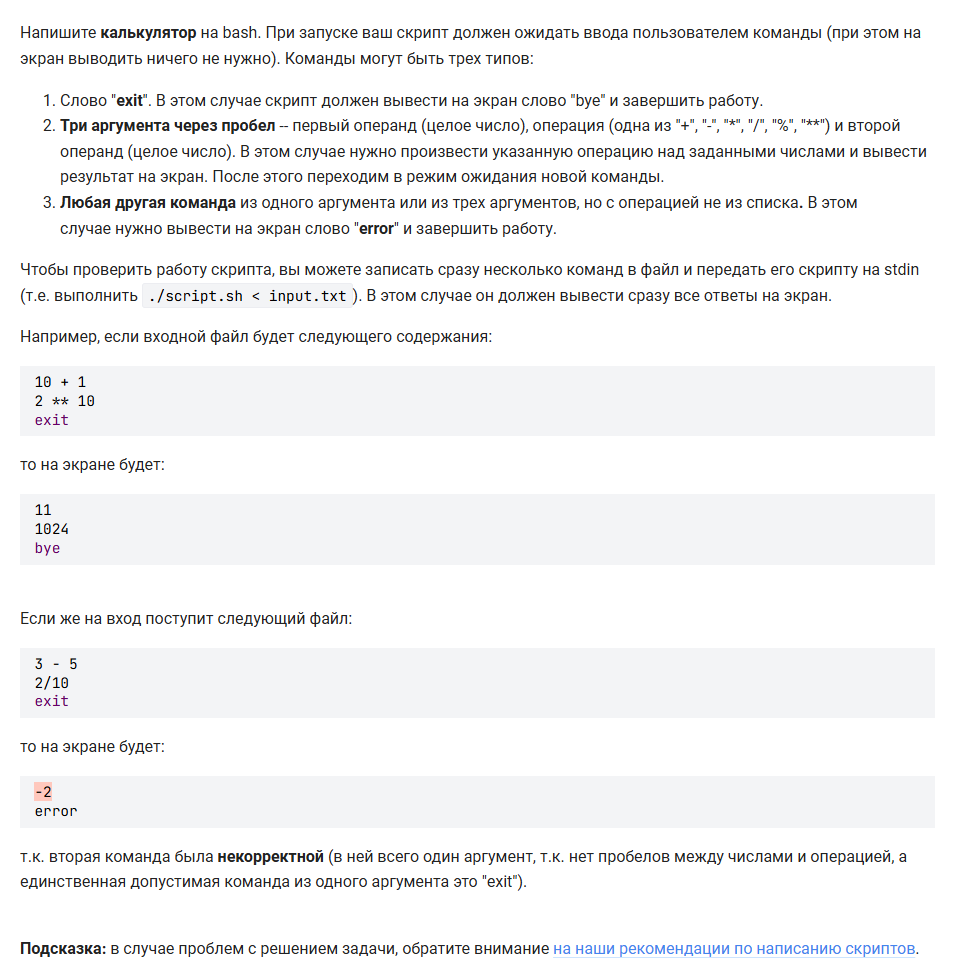


Рис. 23: Задание №20. Условие

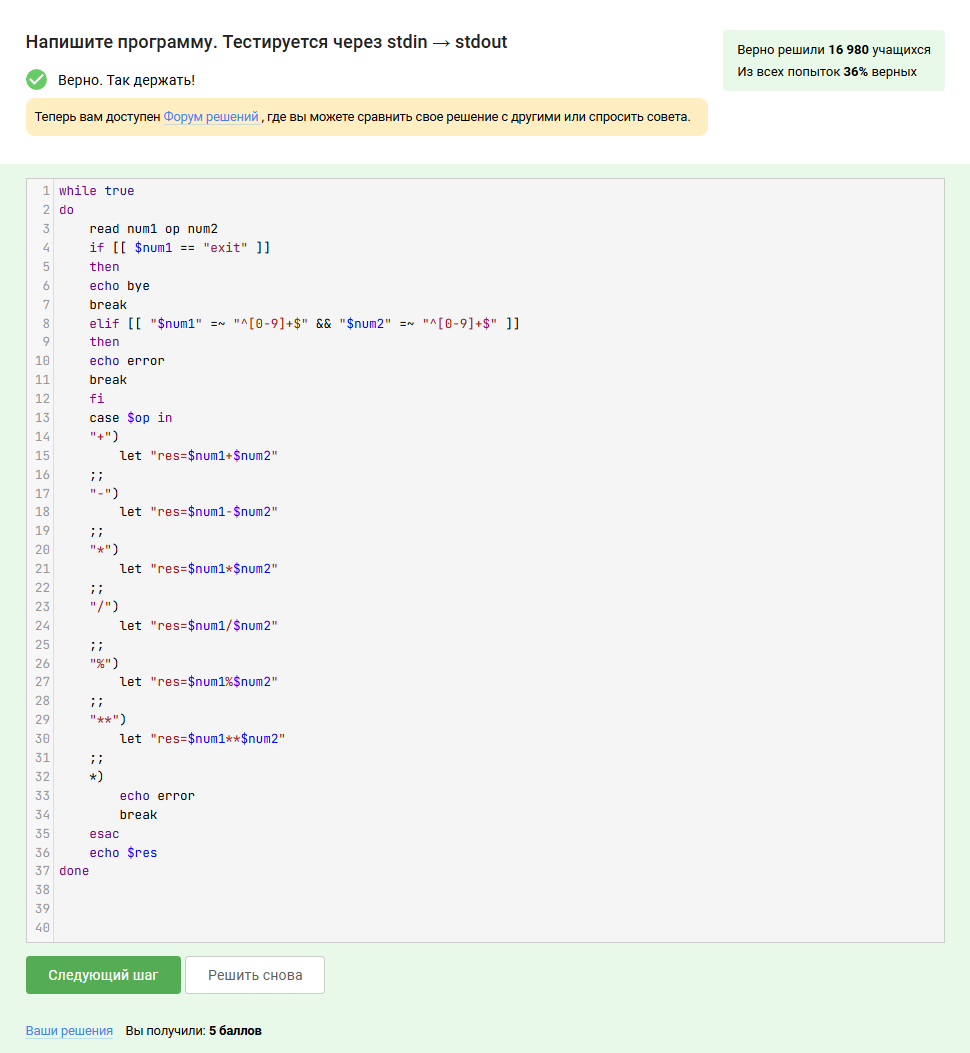


Рис. 24: Задание №20. Ответ

**Листинг №3.4.2. Скрипт калькулятора.**

while true  
do  
 read num1 op num2  
 if [[ $num1 == "exit" ]]  
 then  
 echo bye  
 break  
 elif [[ "$num1" =~ "^[0-9]+$" && "$num2" =~ "^[0-9]+$" ]]  
 then  
 echo error  
 break  
 fi  
 case $op in  
 "+")  
 let "res=$num1+$num2"  
 ;;  
 "-")  
 let "res=$num1-$num2"  
 ;;  
 "\*")  
 let "res=$num1\*$num2"  
 ;;  
 "/")  
 let "res=$num1/$num2"  
 ;;  
 "%")  
 let "res=$num1%$num2"  
 ;;  
 "\*\*")  
 let "res=$num1\*\*$num2"  
 ;;  
 \*)  
 echo error  
 break  
 esac  
 echo $res  
done

## 3.5 Продвинутый поиск и редактирование

1. Вопрос 21-й: Команда -iname не учитывает регистр, поэтому найдутся те варианты, которые начинаются с star и заканчиваются любым набором в любом регистре.

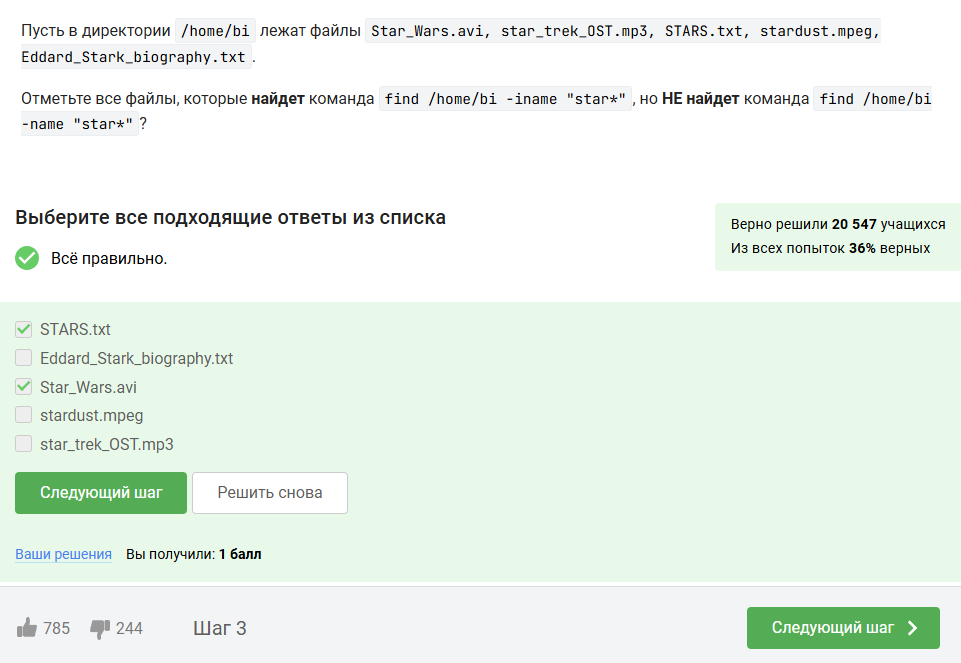


Рис. 25: Задание №21. Условие и ответ

1. Вопрос 22-й: Изучим каждый вариант ответа и выберем верные.

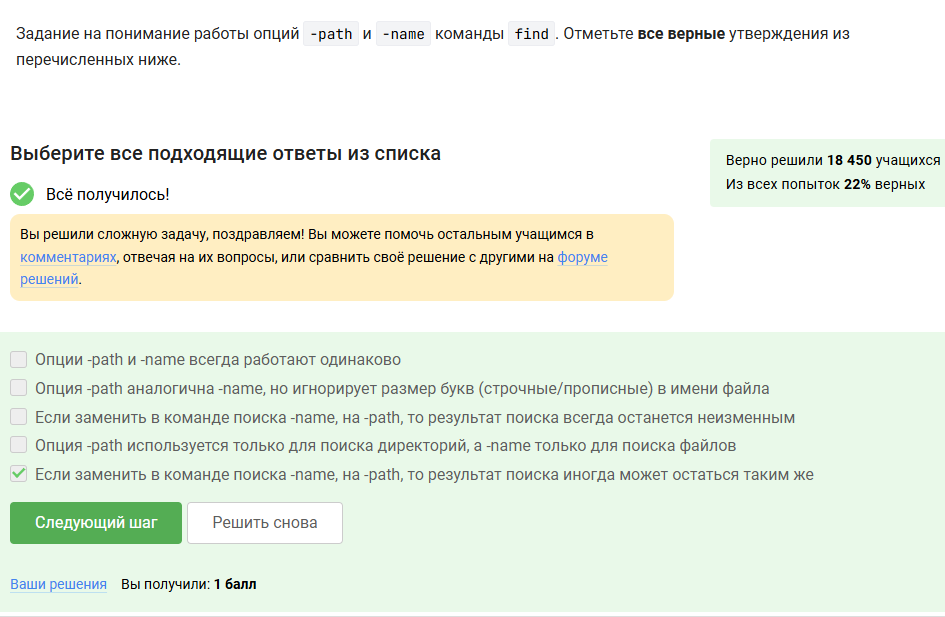


Рис. 26: Задание №22. Условие и ответ

1. Вопрос 23-й: Глубина поиска начинается с текущей директории, поэтому он будет проходить в области dir1 <-> dir2, не заходя в каталог dir3.

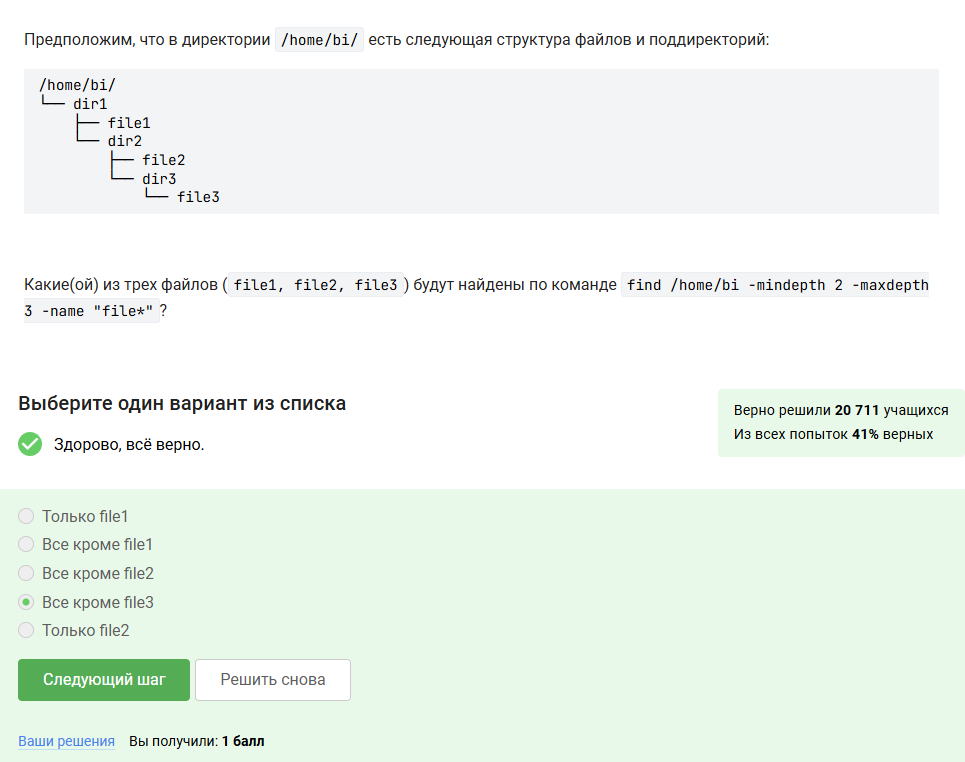


Рис. 27: Задание №23. Условие и ответ

1. Вопрос 24-й: Рассмотрим каждую из опций:

* -A n – Вывод n строк контекста после группы найденных вхождений;
* -B n – Вывод n строк контекста перез группой найденных вхождений;
* -C n – Вывод n строк контекста до и после группы найденных вхождений;

Так как до и после 10 строк слов word нет текста, то контекста не будет, а размер всех файлов будет одинаковый.

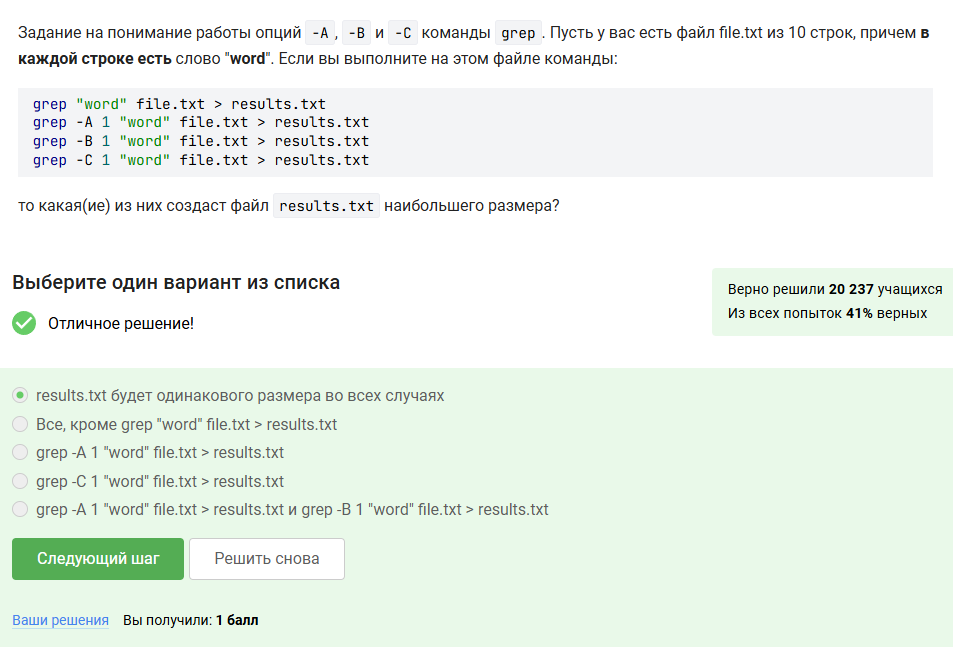


Рис. 28: Задание №24. Условие и ответ

1. Вопрос 25-й: Команда выведет только строки, которые: заканчиваются на слове Ubuntu или ubuntu, перед которым слитно находится не более одного вхождения символов из набора xklXKL.

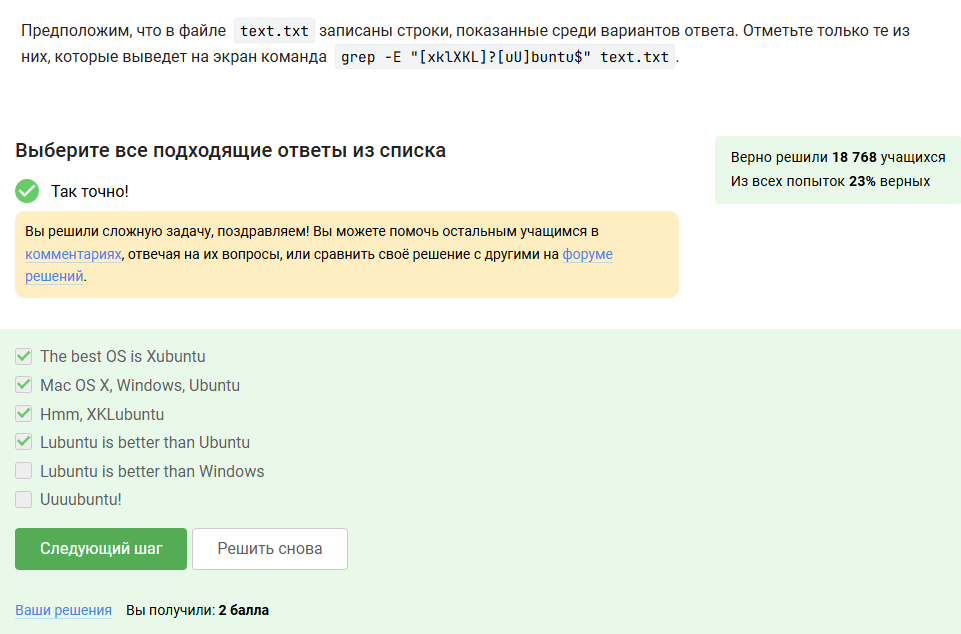


Рис. 29: Задание №25. Условие и ответ

1. Вопрос 26-й: Без опции -n происходит автоматический вывод всех строк, при этом p в конце выражения указывает на то, что дополнительно должны быть выведены строки, удовлетворяющие шаблону. Произойдет задвоение строк.

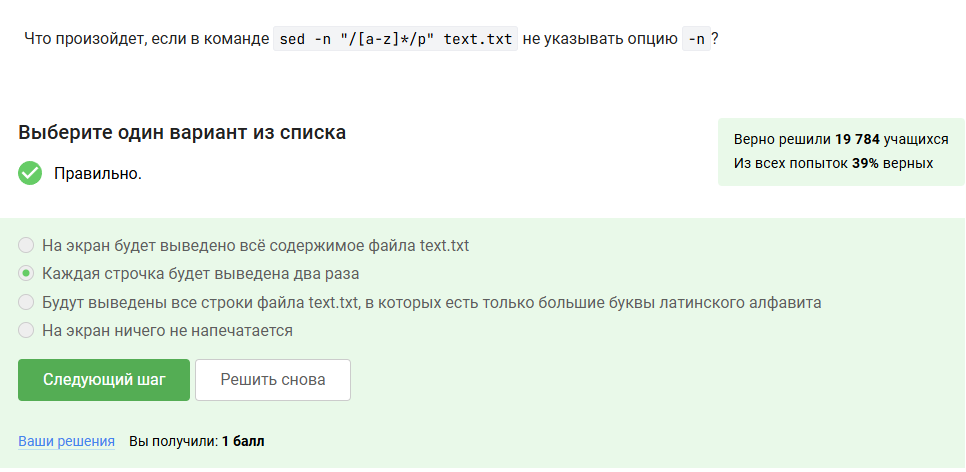


Рис. 30: Задание №26. Условие и ответ

1. Вопрос 27-й: Используем команду sed "s/ [A-Z]\{2,} / abbreviation /g" input.txt > edited.txt, в которой:

* s – указание на копирование;
* [A-Z] – любые символы из набора от A до Z;
* \{2,} – хотя бы две буквы из набора;
* abbreviation – то на что заменяем;
* g – каждое вхождение в строке;

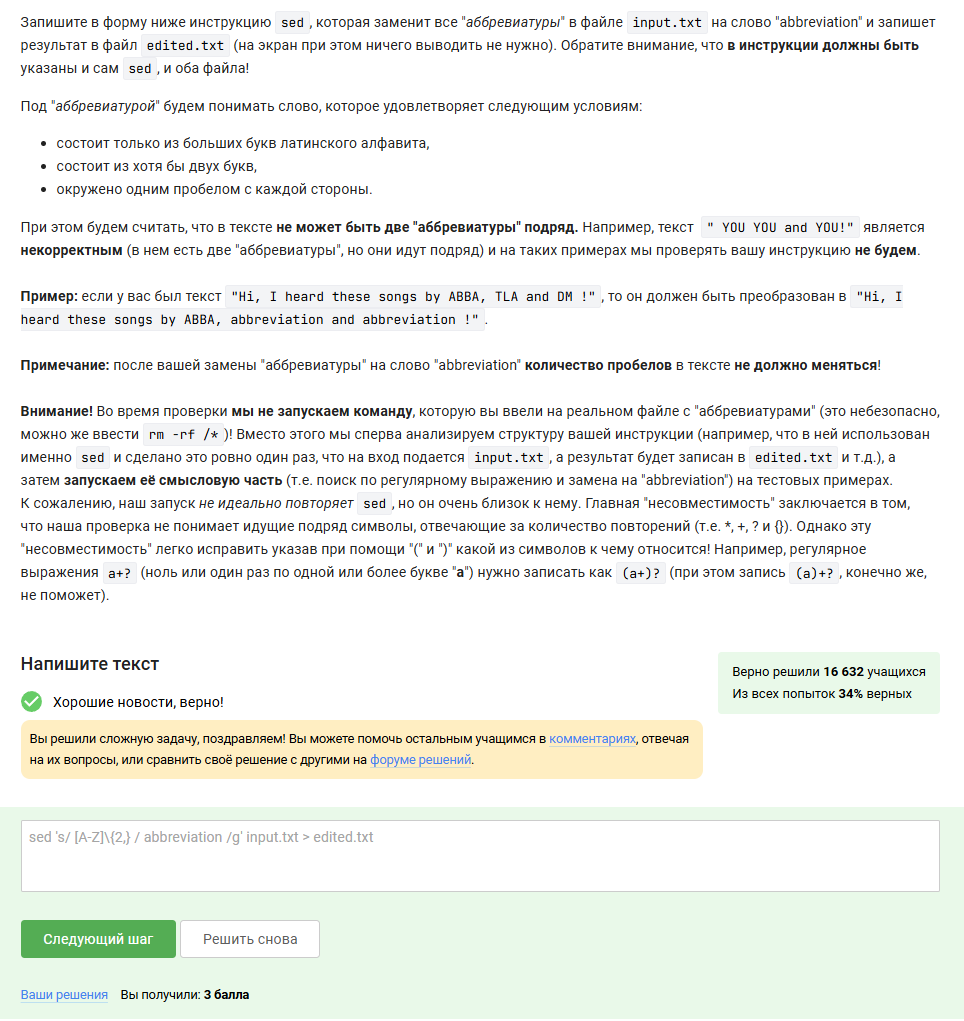


Рис. 31: Задание №27. Условие и ответ

## 3.6 Строим графики в gnuplot

1. Вопрос 28-й: Изучим справку man gnuplot. Из неё получим, что существует ключ -p или же --persist, который позволяет оставить окно графика после закрытия терминала.

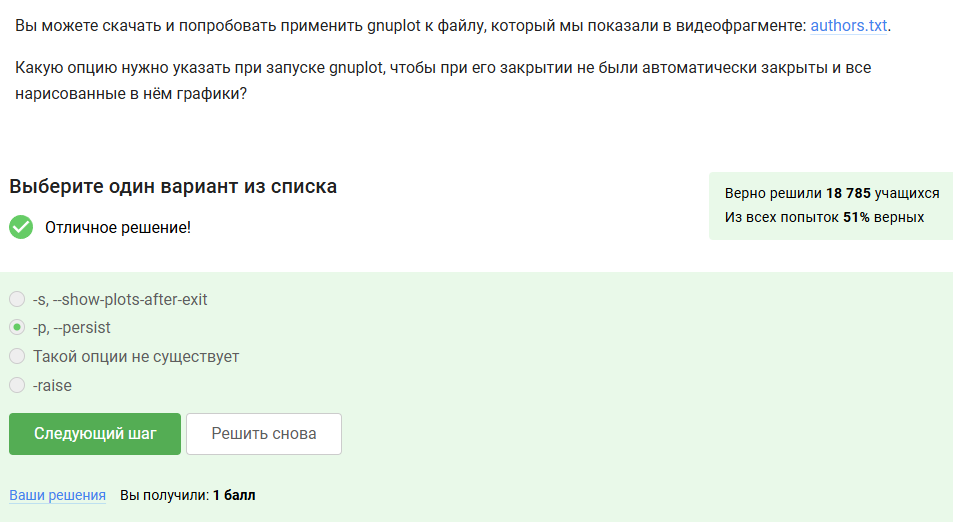


Рис. 32: Задание №28. Условие и ответ

1. Вопрос 29-й: Первая строка указывает gnuplot использовать первую строку файла как название столбцов, так как файлы названий столбцов отсутствуют, gnuplot присвоит ряду стандартное название во второй строке – 'data.csv' using 1:2.

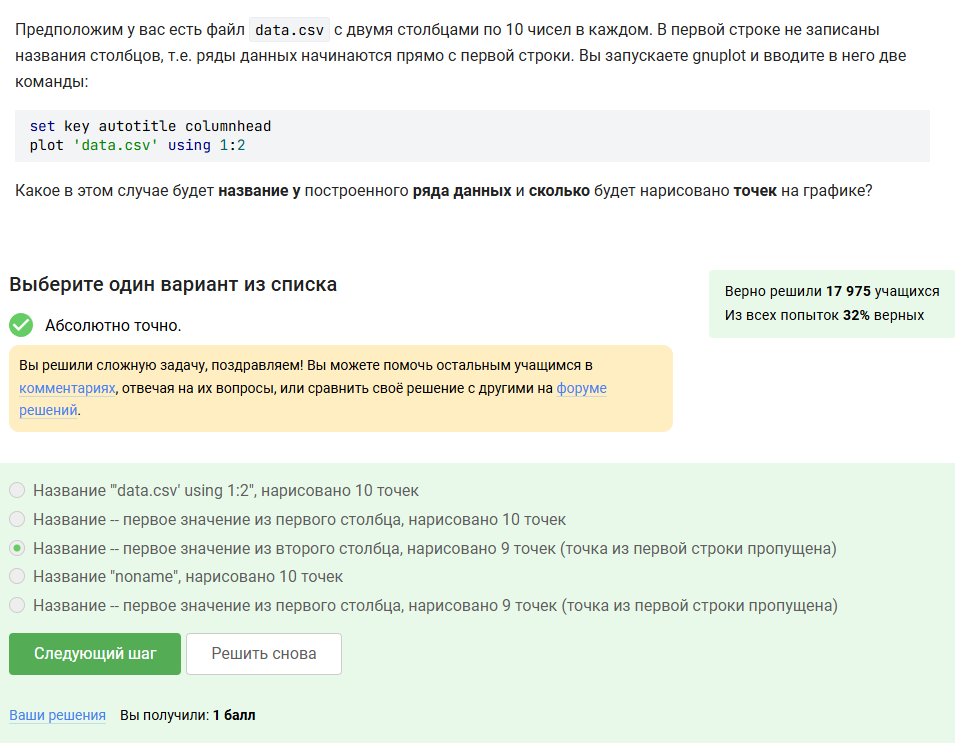


Рис. 33: Задание №29. Условие и ответ

1. Вопрос 30-й: Установим деления командой set, укажем ось Ox с помощью xics и введем через запятую точки в формате (“название”, значение).

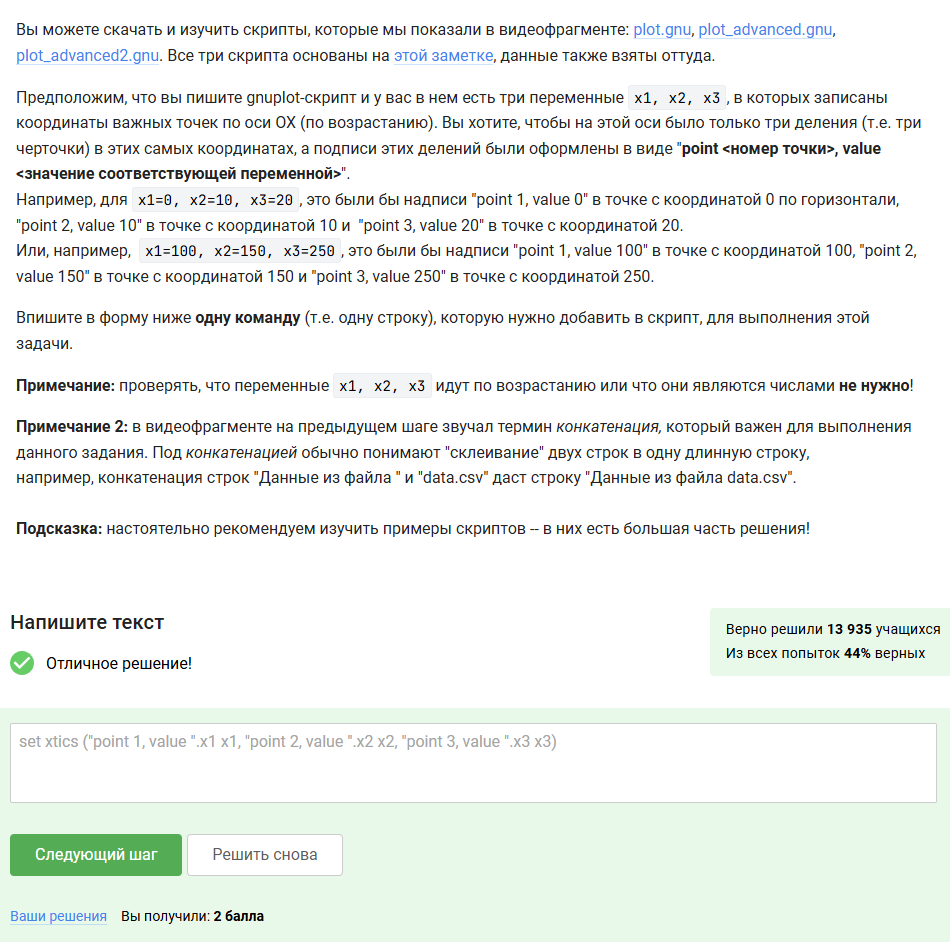


Рис. 34: Задание №30. Условие и ответ

1. Вопрос 31-й: В данном случае график:

* отразится зеркально, если поставить отрицание перед функцией и раскрыть скобки: splot -x\*\*2-y\*\*2
* станет вращаться в обратную сторону, если поменять с 10 до 350 градусов: zrot=(zrot+350)%360
* станет вращаться в два раза быстрее, если уменьшить задержку в два раза: pause 0.1

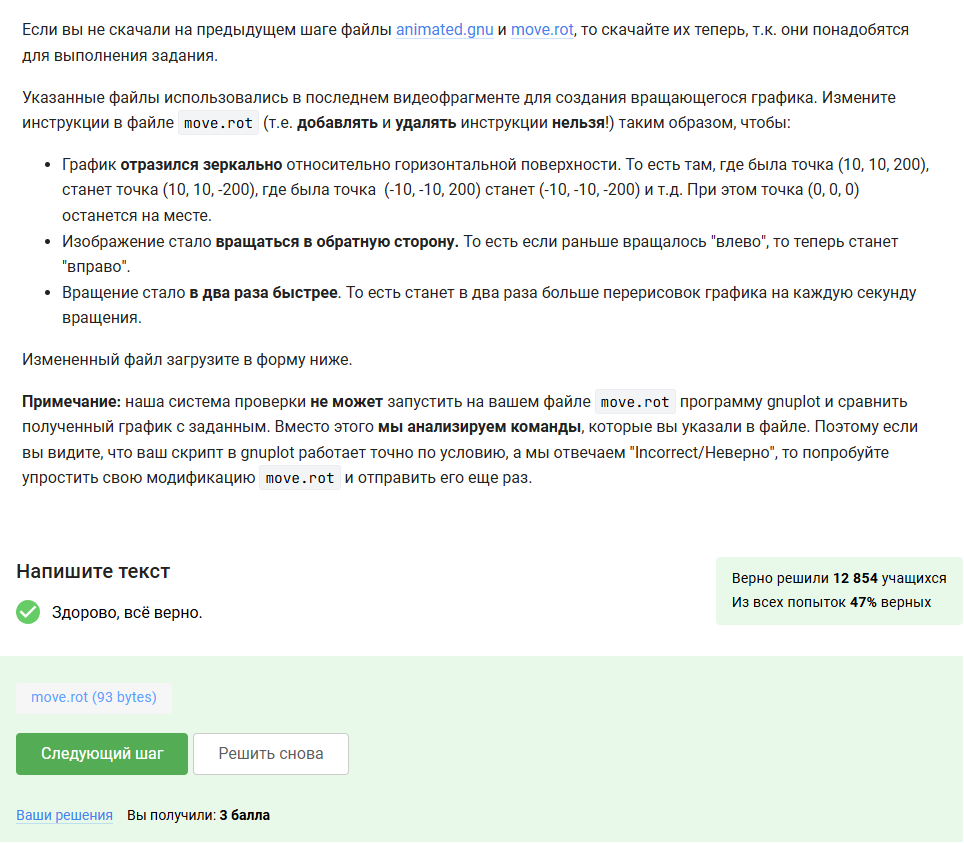


Рис. 35: Задание №31. Условие и ответ

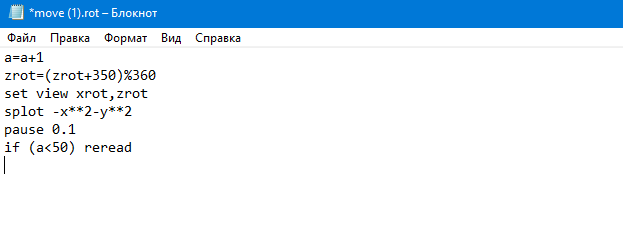


Рис. 36: Задание №31. Файл

## 3.7 Разное

1. Вопрос 32-й: Проверим каждую строку на правильность изменения прав файла и выберем нужные варианты ответа.

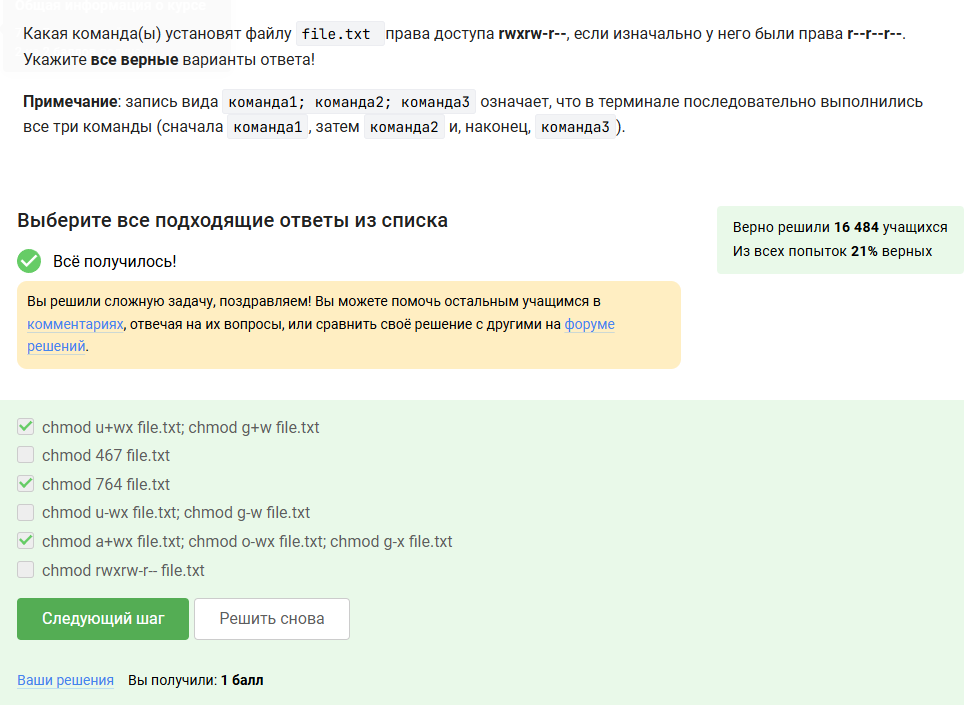


Рис. 37: Задание №32. Условие и ответ

1. Вопрос 33-й: Так как изначальные права у root, то владелец с помощью sudo должен дать права пользователю. Без sudo права не поменяются. Выберем верные варианты ответа.



Рис. 38: Задание №33. Условие

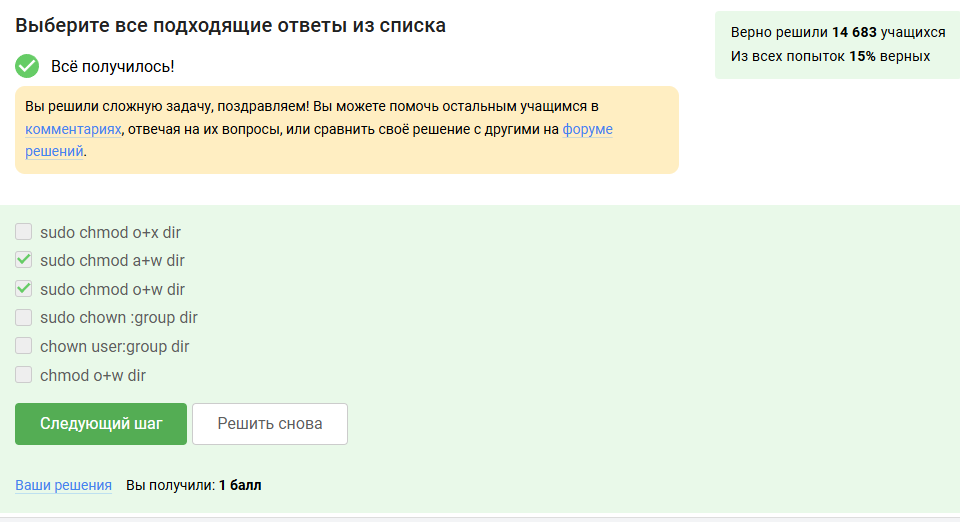


Рис. 39: Задание №33. Ответ

1. Вопрос 34-й: Откроем справку по команде wc и просмотрим ключи:

* -l имя\_файла – вывести количество строк;
* -c имя\_файла – вывести количество байт;
* -m имя\_файла – вывести количество символов;
* -L имя\_файла – вывести длину самой длинной строки;
* -w имя\_файла – вывести количество слов;

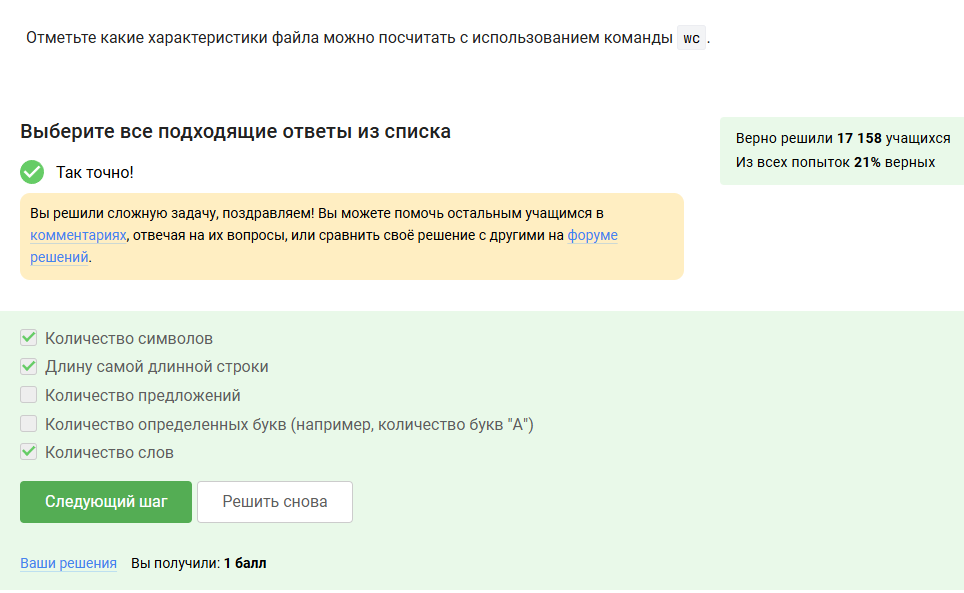


Рис. 40: Задание №34. Условие и ответ

1. Вопрос 35-й: Выведем команду du, анализирующую занимаемое файловое пространство, с двумя ключами -h или --human-readable и -s или --summarize.

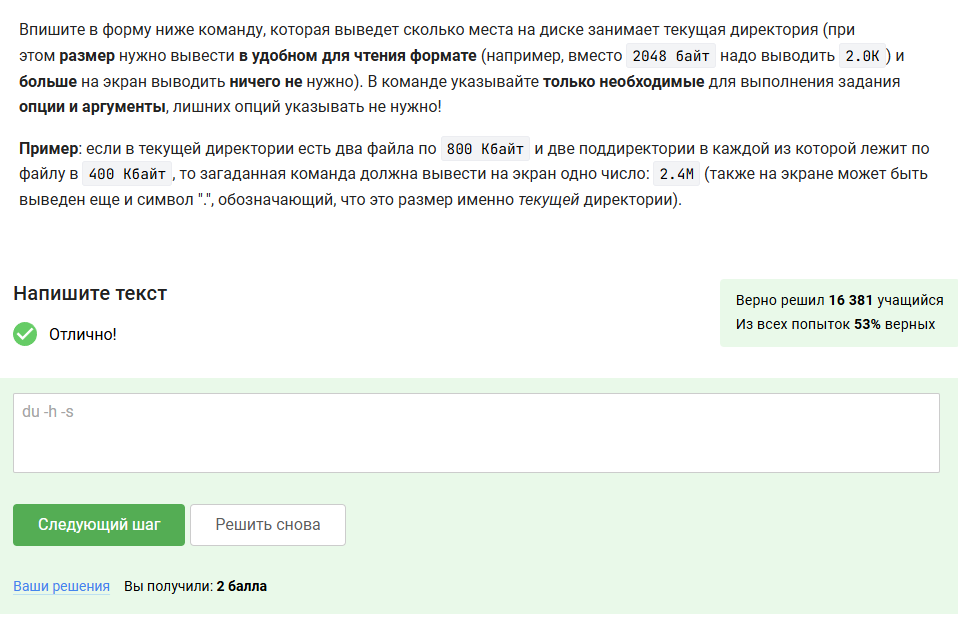


Рис. 41: Задание №35. Условие и ответ

1. Вопрос 36-й: Создадим директорию с помощью, а номера укажем с помощью команды {1..3}, которая проходит по значениям от 1 до 3.

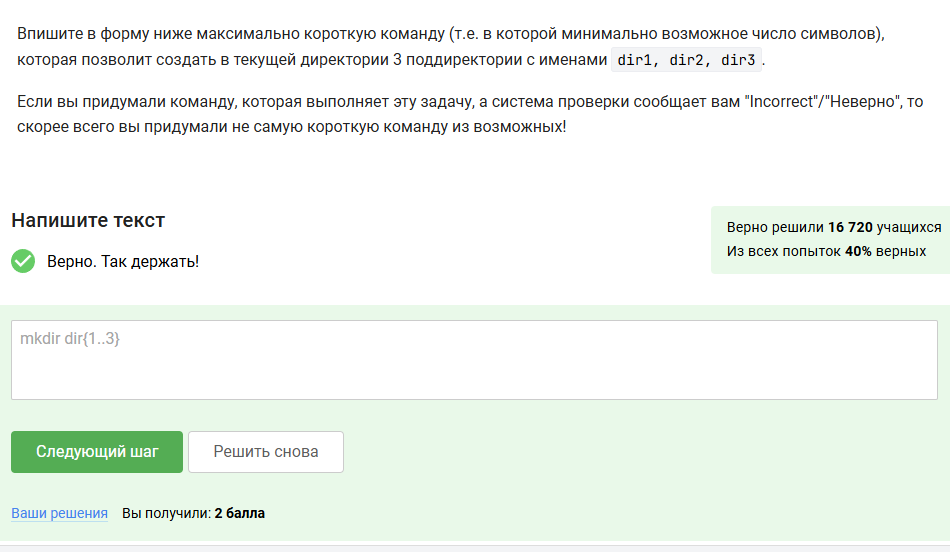


Рис. 42: Задание №36. Условие и ответ

# 4 Вывод

В результате выполнения блока внешнего курса №3 “Продвинутые темы” я ознакомился с редактором vim и написанием скриптов на bash, изучил продвинутый поиск файлов и их продвинутое редактирование и научился строить графики в gnuplot.

# Список литературы

1. Gurevich A., Przhibelskiy A. Введение в Linux [Электронный ресурс]. Stepik. URL: <https://stepik.org/course/73/syllabus>.