Отчет о прохождении 3 этапа внешних курсов

Продвинутые темы

Барето Вилиан Мануел

Содержание

# 1 Цель работы

Ознакомиться с функционалом операционной системы Linux.

# 2 Задание

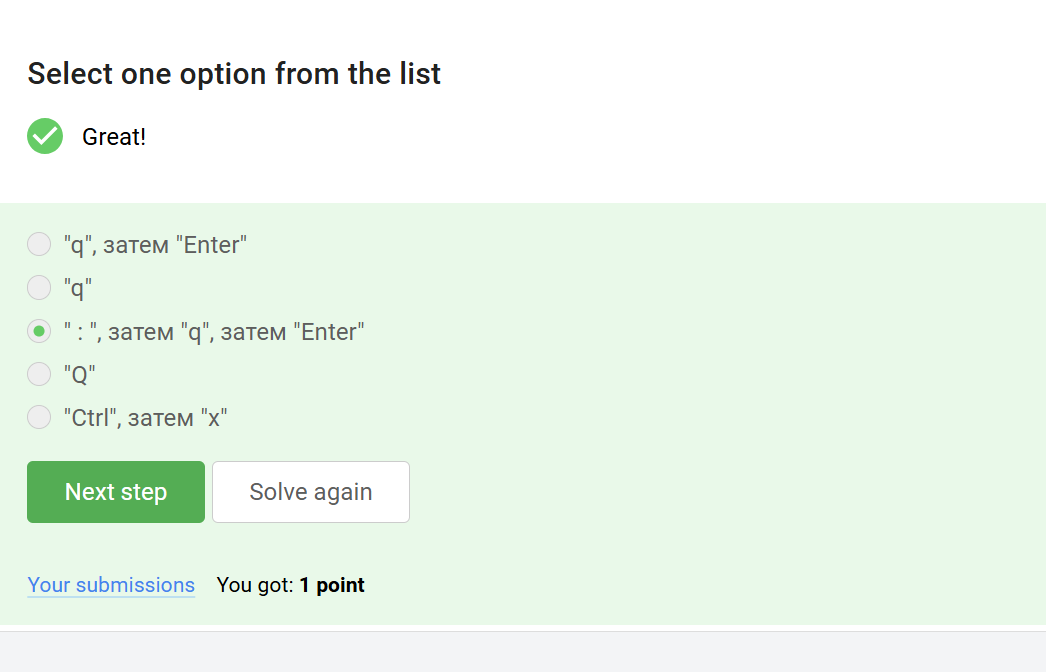
Просмотреть видео и на основе полученной информации пройти тестовые задания.

# 3 Теоретическое введение

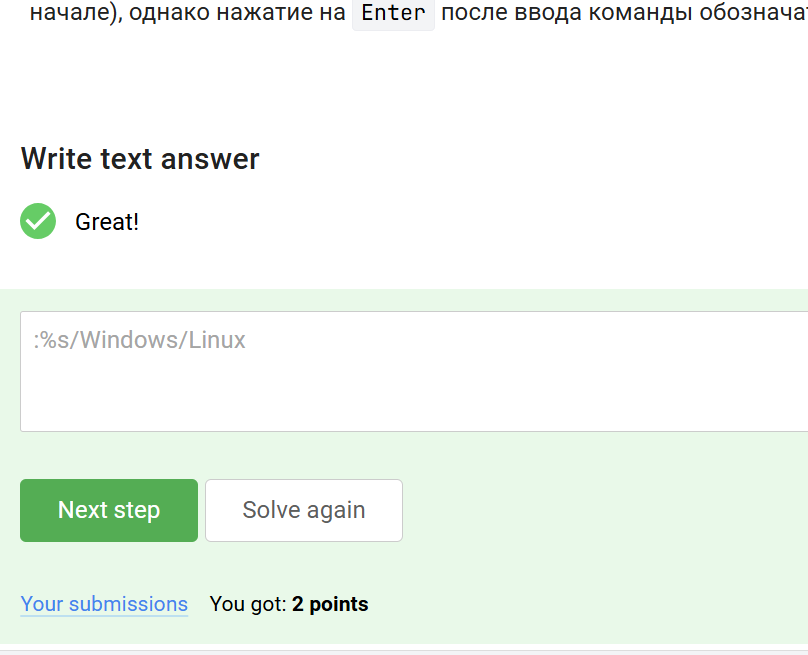
Линукс - в части случаев GNU/Linux — семейство Unix-подобных операционных систем на базе ядра Linux, включающих тот или иной набор утилит и программ проекта GNU, и, возможно, другие компоненты. Как и ядро Linux, системы на его основе, как правило, создаются и распространяются в соответствии с моделью разработки свободного и открытого программного обеспечения. Linux-системы распространяются в основном бесплатно в виде различных дистрибутивов — в форме, готовой для установки и удобной для сопровождения и обновлений, — и имеющих свой набор системных и прикладных компонентов, как свободных, так и проприетарных.

# 4 Выполнение лабораторной работы

3 Этап:

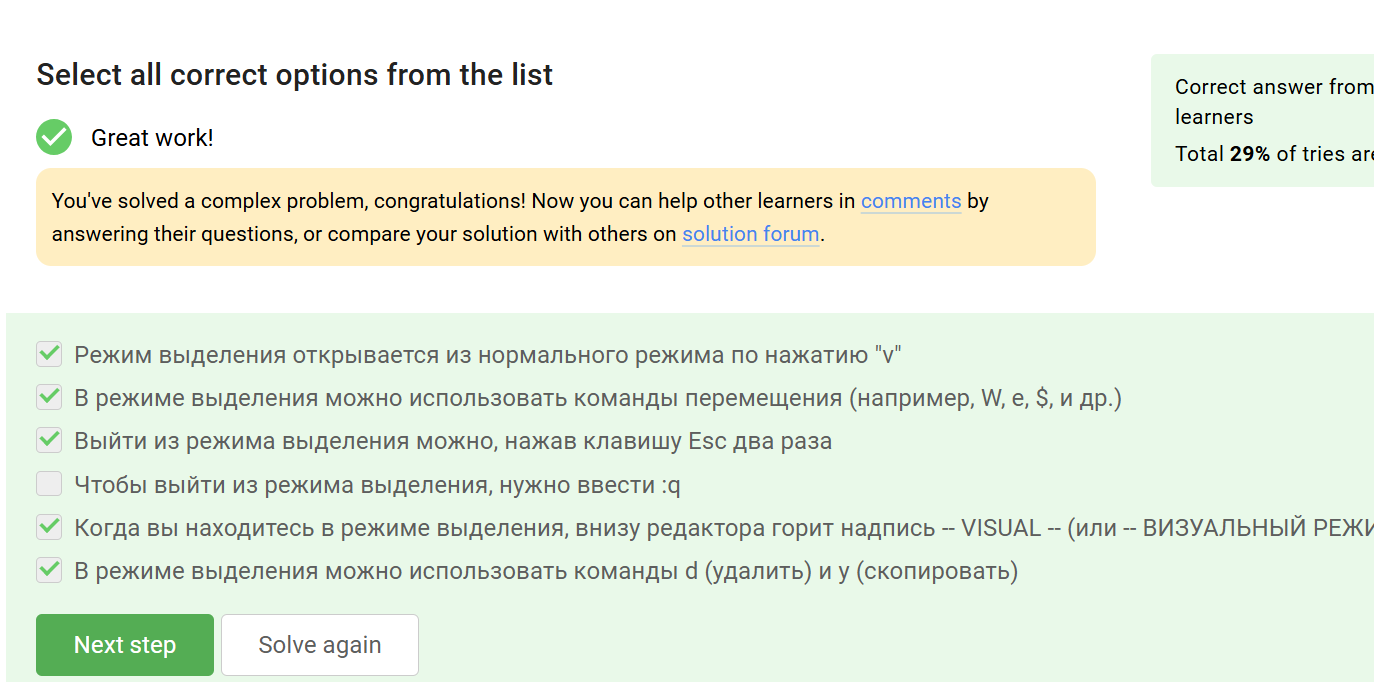


* $ — в конец текущей строки;
* w — на слово вправо;
* b — на слово влево;
* i — начать ввод перед курсором;
* p — вставка содержимого неименнованного буфера под курсором;
* P — вставка содержимого неименованного буфера перед курсором;
* yy (также Y) — копирование текущей строки в неименованный буфер;
* yy — копирование числа строк начиная с текущей в неименованный буфер;



Поиск и замена в редакторе работают по следующей схеме:

Для замены во всем файле можно использовать символ %.



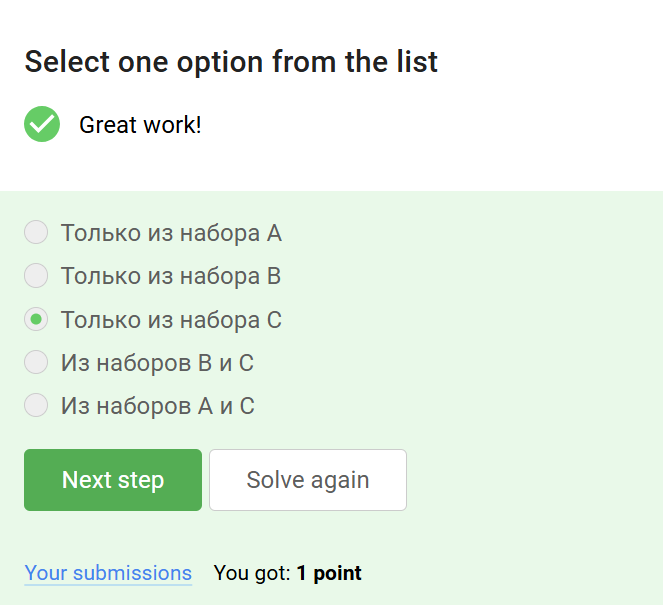
Команда $ — в конец текущей строки, W - до пробела вправо - то есть, перемещение.

Нажать Esc достаточно один раз, но да ладно.

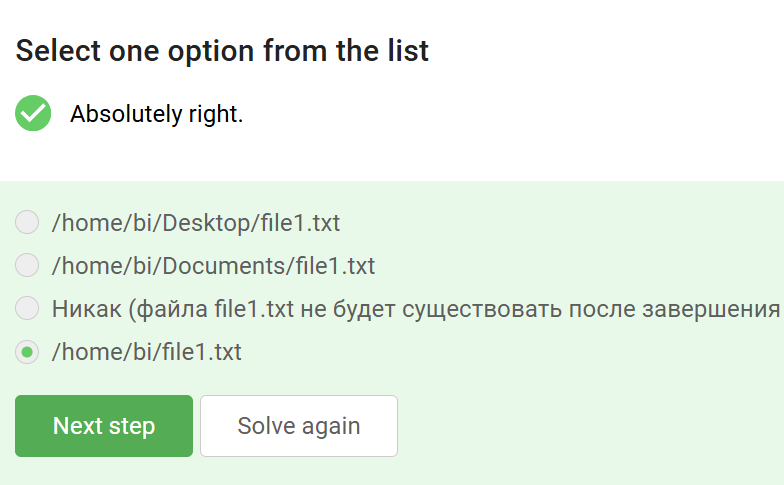
Надпись visual - горит.

d — используется совместно с командами перемещения. Удаляет символы с текущего положения курсора до положения после ввода команды перемещения.

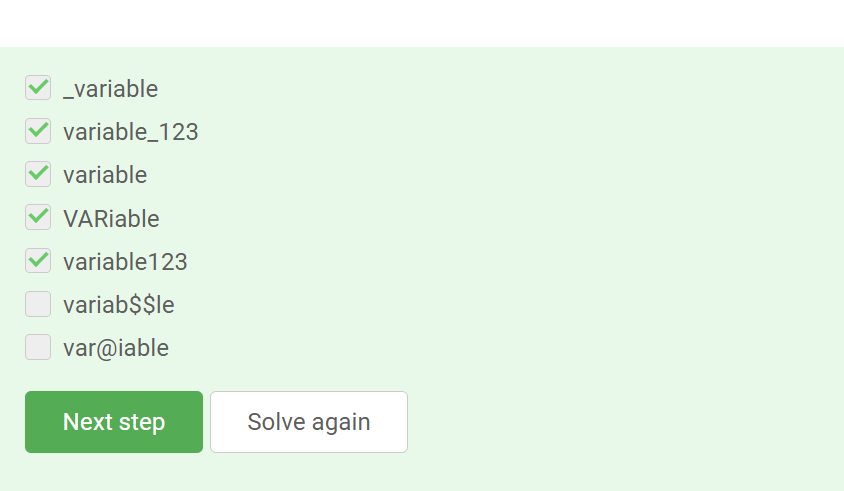
yy (также Y) — копирование текущей строки в буфер;



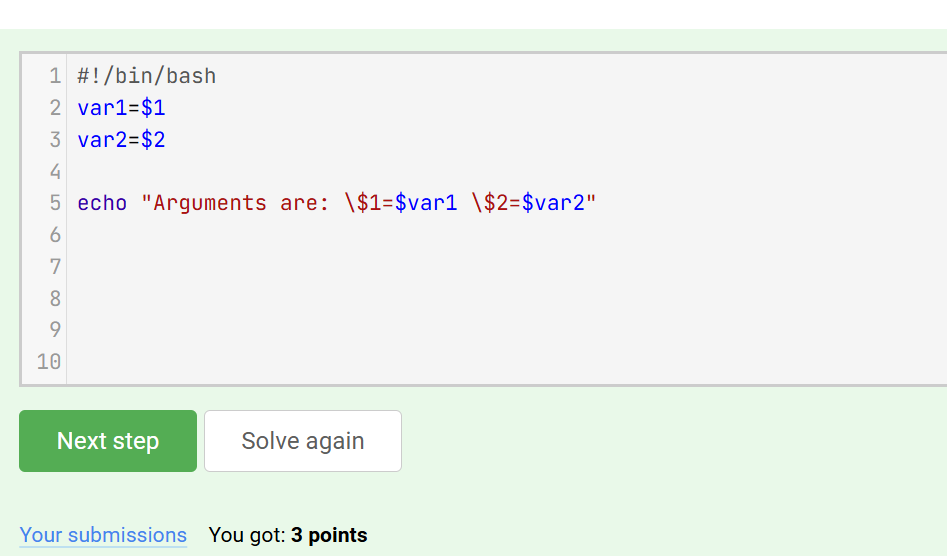
Только из набора С потому что у каждой оболочки свой буфер, который при выходе из нее буде записываться в файл истории.



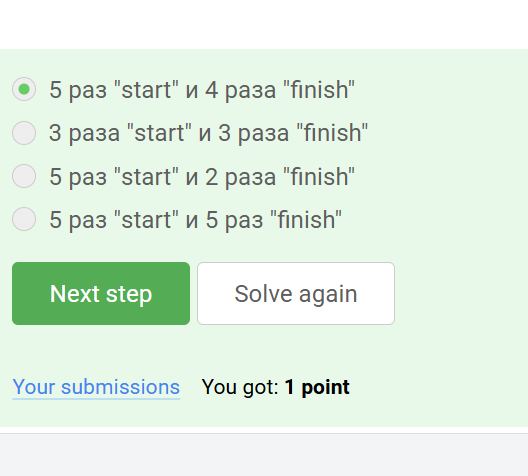
/home/bi/file1.txt - потому что именно в этой директории мы создаем новый файл, а уже после его создания мы переходим в другую папку.



Имя не может начинаться с цифры, содержать специальные символы или пробелы.



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



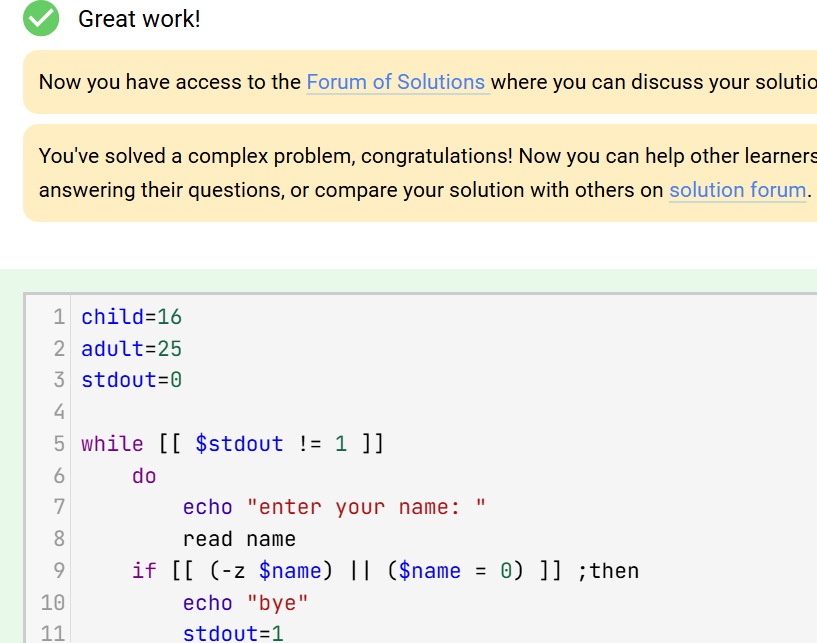
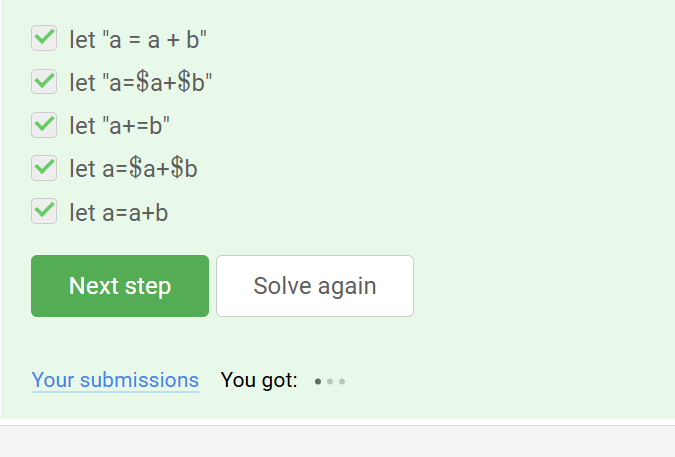
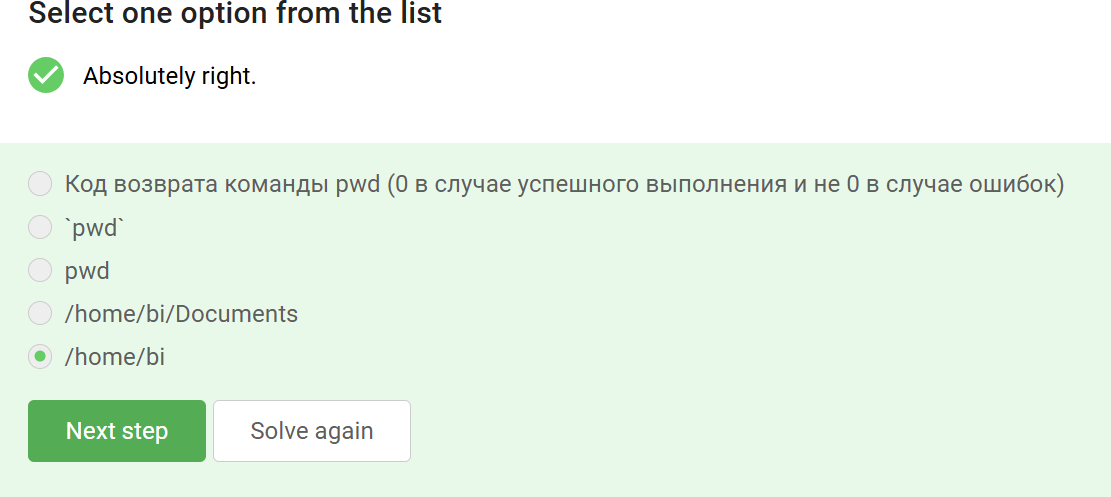


Рис. 1: Задание 14

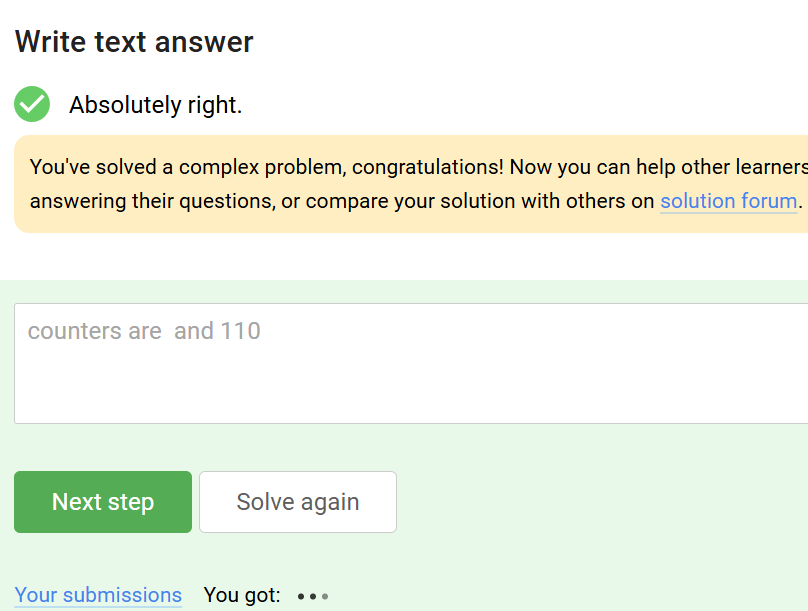
child=16  
adult=25  
stdout=0  
  
while [[ $stdout != 1 ]] #конструкция типа while-True  
 do  
 echo "enter your name: " #Пользователь вводит имя  
 read name  
 if [[ (-z $name) || ($name = 0) ]] ;then #Если имя не по параметрам, простимся  
 echo "bye"  
 stdout=1  
 elif [[ -n $name ]]; then #А вот если имя нормальное  
 while [[ $stdout != 1 ]] ;do   
 echo "enter your age: " #То пусть вводит возраст  
 read age #Считываем возраст  
 if [[ ($age -eq 0) || (-z $age) ]] ;then #Если возраст 0 или строка пуста - прощаемся  
 echo "bye"  
 stdout=1  
 elif [[ $age -le $child ]] ;then #Если меньше или равен ребенку, то ребенок  
 echo "$name, your group is child"  
 elif [[ $age -gt $adult ]] ; then #Больше взрослого - то взрослый  
 echo "$name, your group is adult" ;else  
 if [[ ($age -ge 17) && ($age -le 25) ]] ;then #Если от 17 до 25, то подросток.  
 echo "$name, your group is youth" ;fi  
 fi ;break  
 done ;fi  
done



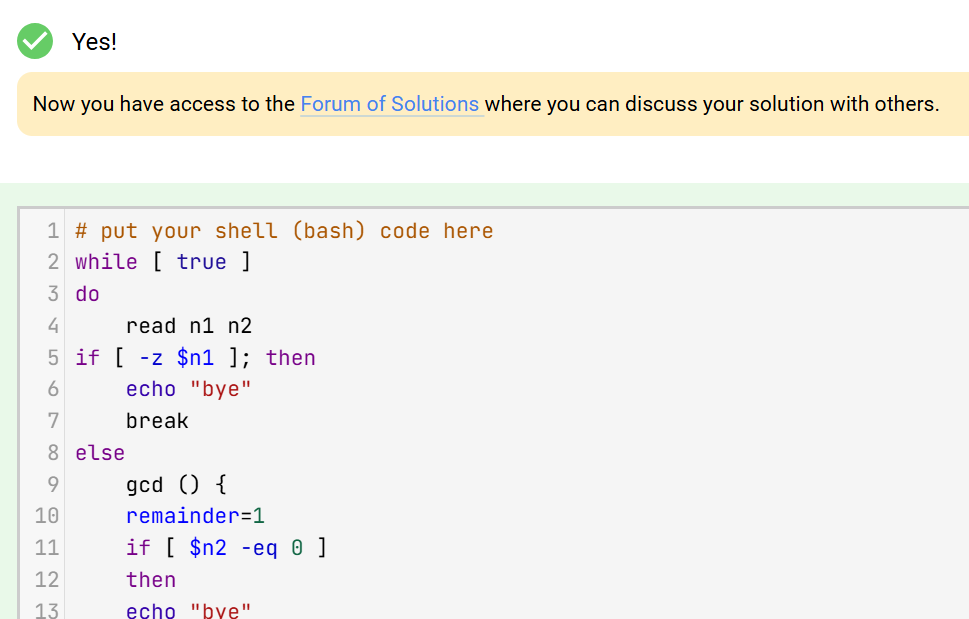
1. a = $a
2. a += b это то же самое, что и a = a + b, но с символами “+=” != “=+”
3. если выражение не в скобках, но с пробелами - работать не будет. (let a=a+b - сработает; let a = a + b - нет)

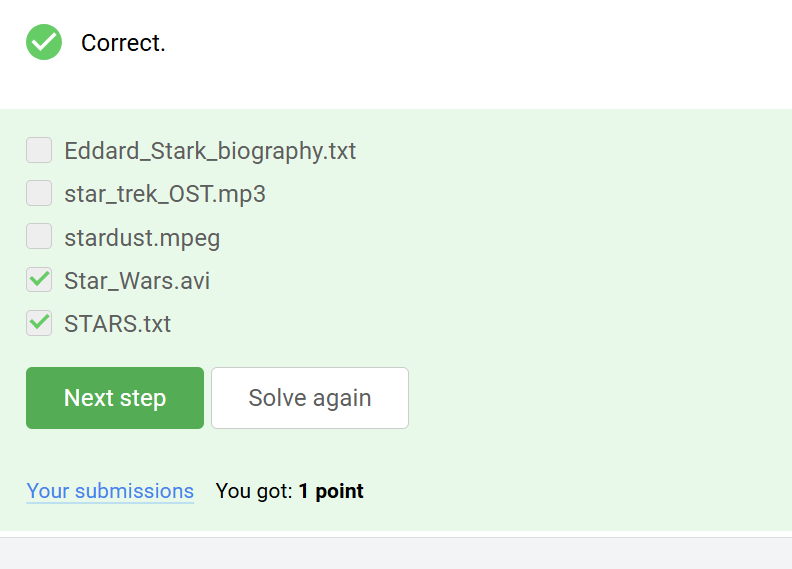


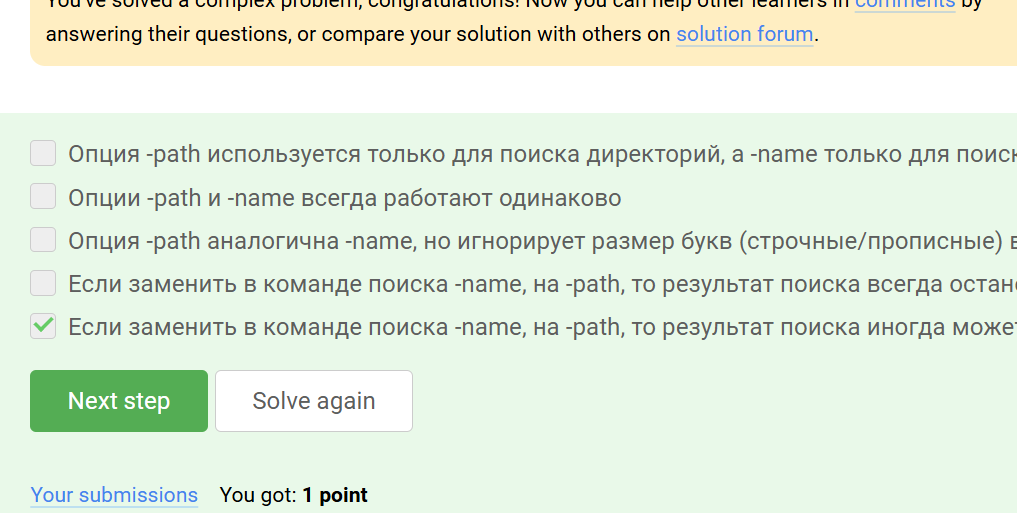
programm выполняет стандартный вывод в терминал (если это принцип работы программы). И нам нужно настроить вывод в файл.



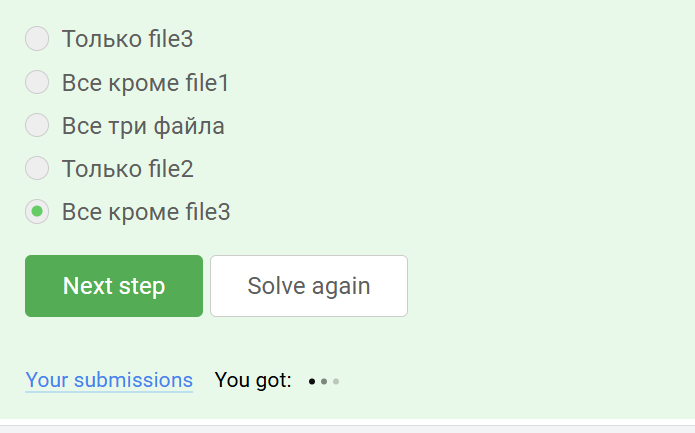
Первая переменная локальная, и это просто пустая строка, вторая переменная - это сумма арифметической прогрессии от 1 до 10, равна 55, но при умножении на 2 даст 110.



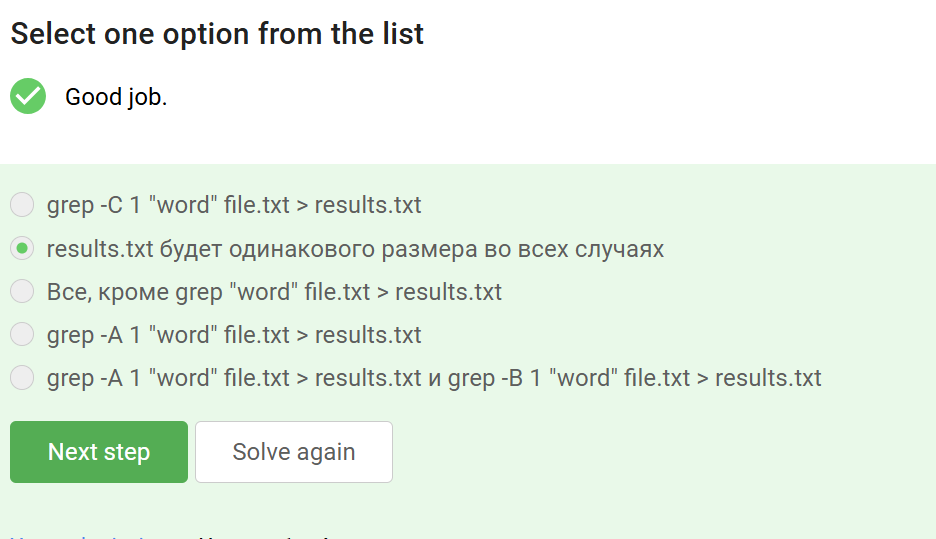




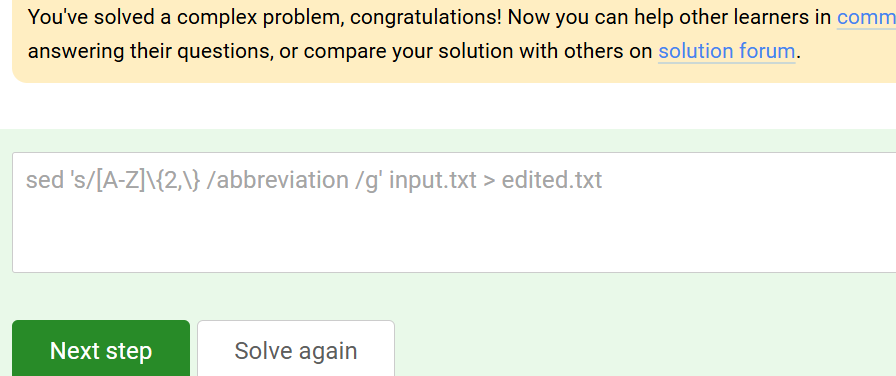
Калькулятор выглядит обычно - мы вводим два числа, пишем, что с ними надо сделать, и потом, учитывая случаи ошибок, выводим результат.



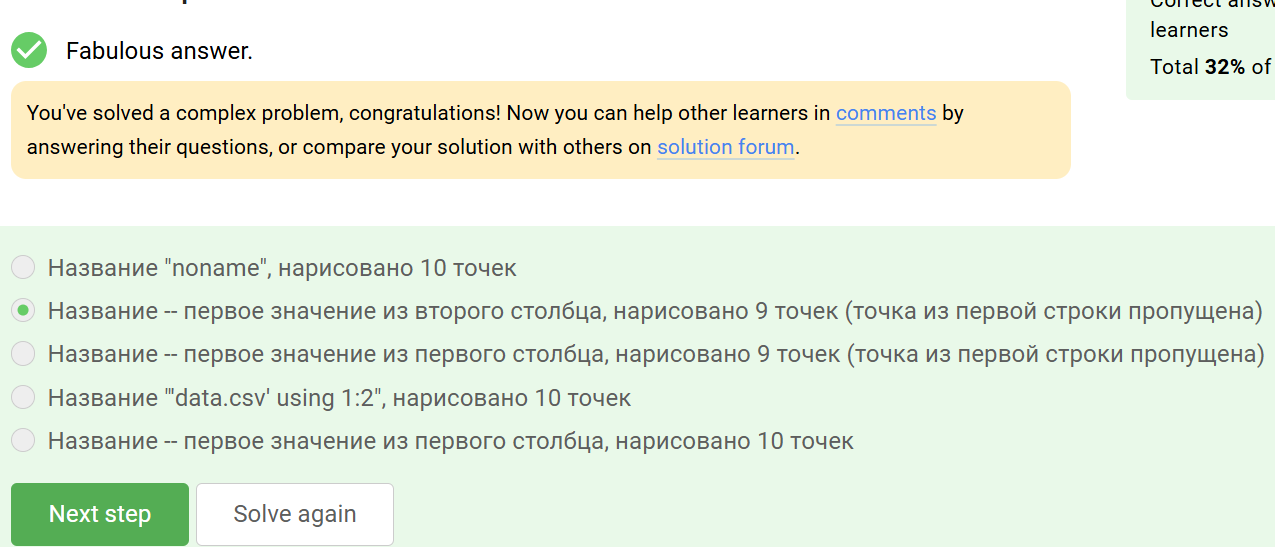
-iname ищет без учета регистра, а -name в точности как в запросе. Звездочка стоит после слова - это значит после слова может быть сколько угодно символов.



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

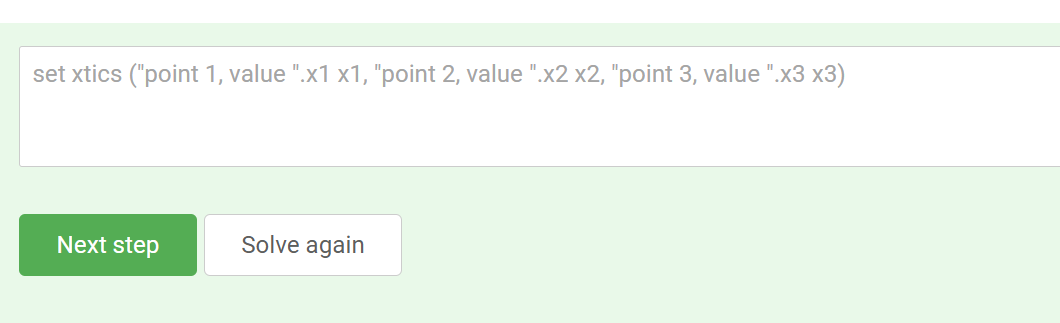


The -n option disables the automatic printing, which means the lines you don’t specifically tell it to print do not get printed, and lines you do explicitly tell it to print (e.g. with p) get printed only once.

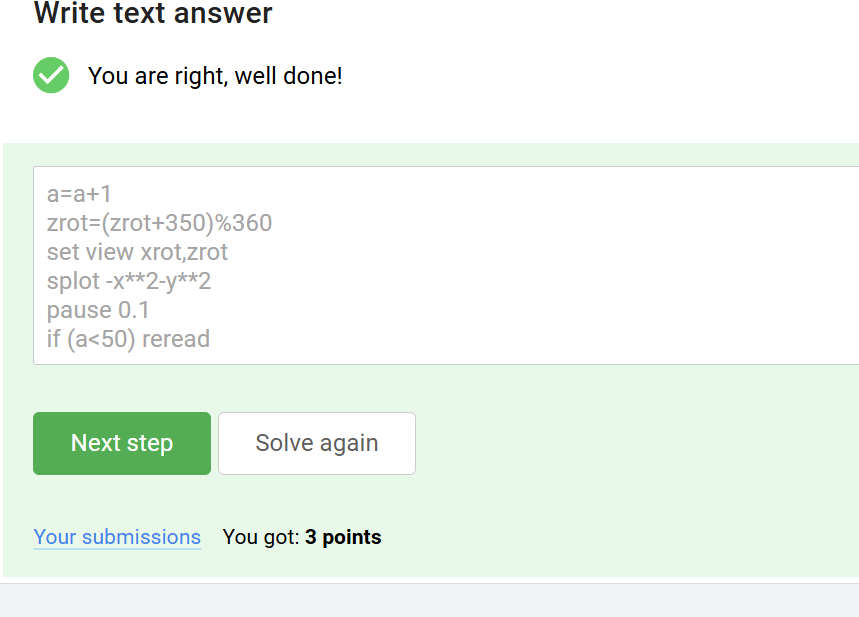


аббревиатура ABBA отличается от двух других аббревиатур тем, что справа он неё стоит запятая без пробела: “ABBA,”.

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.



-persist lets plot windows survive after main gnuplot program exits.



* r - чтение;
* w - запись;
* x - выполнение;
* s - выполнение от имени суперпользователя (дополнительный);
* u - владелец файла;
* g - группа файла;
* o - все остальные пользователи;
* 0 - никаких прав;
* 1 - только выполнение;
* 2 - только запись;
* 3 - выполнение и запись;
* 4 - только чтение;
* 5 - чтение и выполнение;
* 6 - чтение и запись;
* 7 - чтение запись и выполнение.

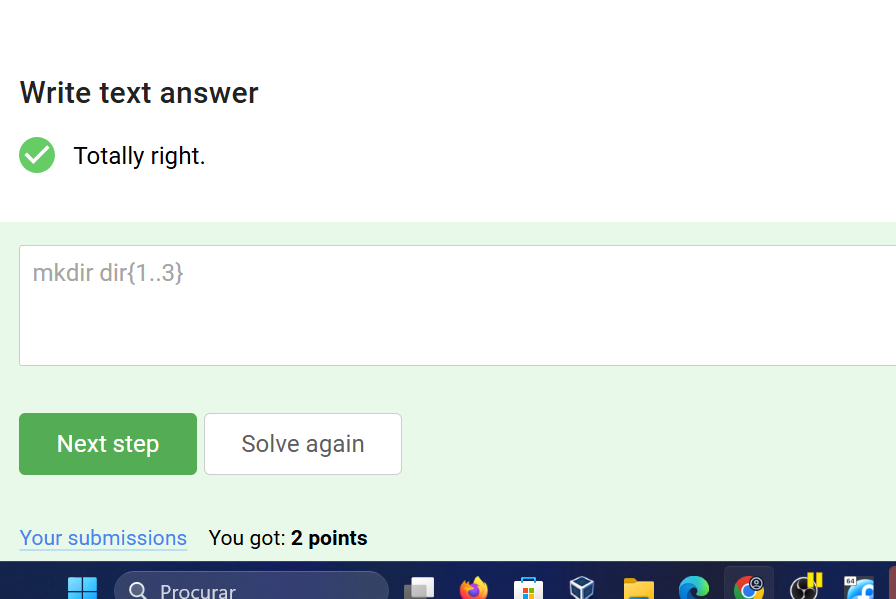


Рис. 2: Задание 32

# 5 Сертификат

|  |
| --- |
| Сертификат |

Рис. 3: Сертификат

# 6 Выводы

Я просмотрела курс и освежила в памяти навыки работы с более сложными командами в Линукс.

# Список литературы

1. Введение в Linux