

# **Внешний курс**

## **Раздел 3**

Полякова Юлия Александровна

# **Содержание**

<b>1 Цель работы</b>	<b>6</b>
<b>2 Задание</b>	<b>7</b>
<b>3 Выполнение этапов курса</b>	<b>8</b>
<b>4 Выводы</b>	<b>43</b>

# Список иллюстраций

3.1 3.1 Шаг 5 . . . . .	8
3.2 3.1 Шаг 7 . . . . .	9
3.3 3.1 Шаг 9 . . . . .	10
3.4 3.1 Шаг 11 . . . . .	11
3.5 3.1 Шаг 12 . . . . .	12
3.6 3.2 Шаг 3 . . . . .	13
3.7 3.2 Шаг 5 . . . . .	14
3.8 3.2 Шаг 7 . . . . .	15
3.9 3.2 Шаг 10, формулировка задания . . . . .	16
3.10 3.2 Шаг 10, листинг . . . . .	16
3.11 3.2 Шаг 10 . . . . .	17
3.12 3.3 Шаг 3 . . . . .	17
3.13 3.3 Шаг 5 . . . . .	18
3.14 3.3 Шаг 6, формулировка задания . . . . .	19
3.15 3.3 Шаг 6, командный файл . . . . .	19
3.16 3.3 Шаг 6, проверка работы . . . . .	20
3.17 3.3 Шаг 8 . . . . .	21
3.18 3.3 Шаг 9, формулировка задания . . . . .	22
3.19 3.3 Шаг 9, командный файл . . . . .	23
3.20 3.3 Шаг 9, проверка работы . . . . .	24
3.21 3.4 Шаг 3 . . . . .	25
3.22 3.4 Шаг 5 . . . . .	26
3.23 3.4 Шаг 6 . . . . .	27
3.24 3.4 Шаг 8 . . . . .	28
3.25 3.4 Шаг 9, формулировка . . . . .	29
3.26 3.4 Шаг 9, командный файл . . . . .	30
3.27 3.4 Шаг 9, проверка работы . . . . .	30
3.28 3.4 Шаг 10, формулировка . . . . .	31
3.29 3.4 Шаг 10, командный файл . . . . .	32
3.30 3.4 Шаг 10, проверка работы . . . . .	33
3.31 3.5 Шаг 3 . . . . .	33
3.32 3.5 Шаг 4 . . . . .	34
3.33 3.5 Шаг 5 . . . . .	35
3.34 3.5 Шаг 7 . . . . .	36
3.35 3.5 Шаг 9 . . . . .	37
3.36 3.5 Шаг 11 . . . . .	37
3.37 3.5 Шаг 12 . . . . .	38

3.38 3.6 Шаг 3 . . . . .	39
3.39 3.6 Шаг 5 . . . . .	39
3.40 3.6 Шаг 7 . . . . .	40
3.41 3.6 Шаг 10, формулировка . . . . .	41
3.42 3.6 Шаг 10, измененный файл . . . . .	42

# **Список таблиц**

# **1 Цель работы**

Получить базовые знания по работе с Linux с помощью внешнего курса на stepik.

## **2 Задание**

Пройти третий раздел “Продвинутые темы”. Выполнить тестовые и интерактивные задания.

### 3 Выполнение этапов курса

1. Часть “Текстовый редактор vim”. Здесь представлены 5 заданий.

Шаг 5 - (рис. 3.1). : - переход в режим команд, q - команда выхода, Enter - отправить команду выхода.

3.1 Текстовый редактор vim 12 из 12 шагов пройдено 7 из 7 баллов получено

Какую клавишу(и) нужно нажать на клавиатуре, чтобы выйти из редактора vim? Считайте, что вы только что открыли файл и вам сразу понадобилось выйти из редактора.

Выберите один вариант из списка

Хорошие новости, верно!

Верно решили 32 523 учащихся  
Из всех попыток 69% верных

"Q"  
"Ctrl", затем "X"  
"q"  
 ".," , затем "q", затем "Enter"  
".," , затем "q"

Следующий шаг Решить снова

Ваши решения Вы получили: 1 балл

Рис. 3.1: 3.1 Шаг 5

Шаг 7 - (рис. 3.2). Логично, что W перемещается по большим словам, а w - с меньшей дистанцией, поэтому сразу логически понятны все верные ответы.

3.1 Текстовый редактор vim 12 из 12 шагов пройдено 7 из 7 баллов получено

⊕ 0 **КОП 1**

При перемещении в vim "по словам" есть небольшая разница в том, используем мы маленькую (`w`, `e`, `b`) или большую (`W`, `E`, `B`) букву. Первые перемещают нас по "словам" (`word`), а вторые по "большим словам" (`WORD`). Посмотрите справку по этим перемещениям и разберитесь в чем заключается разница между `word` и `WORD`.

А для того, чтобы убедиться, что вы разобрались, отметьте ниже **все верные** утверждения про следующую строку:  
`Strange_ TEXT is_here. 2=2 YES!`

**Примечание:** во всех утверждениях имеется ввиду, что мы находимся в редакторе vim, включен нормальный режим работы и курсор находится в самом начале строки.

**Подсказка:** чтобы вызвать vim-справку по, например, перемещению `w`, нужно открыть vim и ввести команду `:help w`. Вы попадете в то место справки, где описано это перемещение, а так как все перемещения описаны рядом, то двигаясь по тексту вверх и вниз можно прочитать и про `w` и про `b` и, самое главное, про `word` и `WORD`. Кроме того, можно вызвать сразу справку по термину `word` при помощи `:help word`. Чтобы закрыть справку, нужно ввести команду `:q`.

Выберите все подходящие ответы из списка

Всё правильно.

Верно решили 25 385 учащихся  
Из всех попыток 20% верных

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

- Чтобы попасть в конец строки, нужно совершить меньше нажатий на `W`, чем на `w`
- В этой строке 9 "слов" (`word`)
- В этой строке 9 "больших слов" (`WORD`)
- В этой строке 5 "слов" (`word`)
- Нажимая только на `W`, нельзя переместить курсор на `"`.

Рис. 3.2: 3.1 Шаг 7

Шаг 9 - (рис. 3.3). Проанализировав каждую команду, можно понять, какие из них позволяют добиться этой строчки.

The screenshot shows a user interface for an online challenge. At the top, there's a navigation bar with icons for back, forward, search, and help, followed by a progress indicator showing 0 solved steps out of 12 total, and a 'Join' button.

Below the bar, a message reads: "3.1 Текстовый редактор vim 12 из 12 шагов пройдено 7 из 7 баллов получено".

A task description follows: "Предположим, что в текстовом файле записана одна единственная строка:  
one two three four five  
и вам нужно преобразовать её в строку  
three four four four five".

A note below says: "Какие(ой) из предложенных ниже наборов нажатий клавиш выполняют такое редактирование? В этих наборах нажатие на клавишу Esc обозначается как <Esc> (т.е. знаки "<" и ">" не несут отдельного смысла)."

A note below states: "Примечание: во всех утверждениях имеется в виду, что мы находимся в редакторе vim, включен нормальный режим работы и курсор находится в самом начале строки."

The main area is titled "Выберите все подходящие ответы из списка". It contains a list of options with checkboxes:

- xxxxwwwywywPp
- d2wwifour four <Esc>
- ddithree four four four five<Esc>
- d2wwywpp
- d2dwywPp
- d2w\$bfour four <Esc>

Below the list are two buttons: "Следующий шаг" (Next step) and "Решить снова" (Solve again).

In the top right corner of the main area, a box displays: "Верно решили 23 655 учащихся" and "Из всех попыток 16% верных".

At the bottom left, a small note says: "Ваше решение: Вы получили 1 балл".

Рис. 3.3: 3.1 Шаг 9

Шаг 11 - (рис. 3.4). Довольно простая команда, заменяющая слова.



3.1 Текстовый редактор vim 12 из 12 шагов пройдено 7 из 7 баллов получено

Предположим, что вы открыли файл в редакторе vim и хотите заменить в этом файле все строки, содержащие слово Windows , на такие же строки, но со словом Linux . Если в какой-то строке слово Windows встречается больше, чем один раз, то заменить на Linux в этой строке нужно **только самое первое** из этих слов.

**Какую команду** нужно ввести для этого в vim? Укажите необходимую команду целиком (т.е. включая ввод ":" в самом начале), однако нажатие на Enter после ввода команды обозначать никак не нужно.

Напишите текст

✓ Верно.

Верно решил 24 631 учащийся  
Из всех попыток 57% верных

:%s/Windows/Linux

Следующий шаг

Решить снова

Ваши решения Вы получили: 2 балла

Рис. 3.4: 3.1 Шаг 11

Шаг 12 - (рис. 3.5). После исследования этого режима в лабораторной работе основного курса сразу удалось определить ответы.



3.1 Текстовый редактор vim 12 из 12 шагов пройдено 7 из 7 баллов получено

Мы совсем не рассказали вам про третий режим работы vim – режим **выделения (Visual)**. Предлагаем вам ознакомиться с ним самостоятельно. Например, это можно сделать во время прохождения упражнений в vimtutor, который мы настоятельно рекомендуем вам для изучения vim!

Чтобы убедиться, что вы разобрались с этим режимом работы, отметьте, пожалуйста, **все верные** утверждения из списка ниже.

**Подсказка:** если вы не хотите проходить vimtutor целиком, то можете открыть его и поиском найти слово "Visual". Вы попадете в задание, прохождение которого будет вполне достаточно, чтобы выполнить это задание.

**Выберите все подходящие ответы из списка**

Правильно.

Верно решили 23 497 учащихся  
Из всех попыток 29% верных

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

- Режим выделения открывается при помощи команды :visual
- Режим выделения открывается из нормального режима по нажатию "v"
- В режиме выделения можно использовать команды перемещения (например, W, e, \$, и др.)
- В режиме выделения можно использовать команды d (удалить) и u (скопировать)
- Выйти из режима выделения можно, нажав клавишу Esc два раза
- Когда вы находитесь в режиме выделения, внизу редактора горит надпись -- VISUAL – (или – ВИЗУАЛЬНЫЙ РЕЖИМ –)

[Следующий шаг](#)

[Решить снова](#)

[Ваши решения](#) Вы получили: 2 балла

Рис. 3.5: 3.1 Шаг 12

2. Часть “Скрипты на bash: основы”. Здесь 4 задания.

Шаг 3 - (рис. 3.6). Самый логичный вариант.

The screenshot shows a user interface for an online course. At the top, there are navigation icons (back, forward, search, etc.) and a progress bar indicating '3.2 Скрипты на bash: основы' (10 из 10 шагов пройдено). Below the progress bar, a message reads: 'Надеемся, что вы разобрались, что одну оболочку (например, sh) можно запустить из другой оболочки (например, из bash).'. A descriptive text follows: 'Предположим, что вы открыли терминал и у вас в нем запущена оболочка bash. Вы набираете в ней команды A1, A2, A3, а затем запускаете оболочку sh. В этой оболочке вы набираете команды B1, B2, B3 и запускаете оболочку bash. И, наконец, в этой последней оболочке вы набираете команды C1, C2, C3. Если теперь вы попробуете при помощи стрелочек вверх/вниз перемещаться по истории набранных команд, то команды из какого набора(ов) будут появляться?'.

**Выберите один вариант из списка**

Верно.

Из наборов А и С  
 Только из набора В  
 Только из набора А  
 Из наборов В и С  
 Только из набора С

**Верно решили 30 266 учащихся**  
Из всех попыток 65% верных

**Следующий шаг**   **Решить снова**

Ваши решения   Вы получили: 1 балл

Рис. 3.6: 3.2 Шаг 3

Шаг 5 - (рис. 3.7). Мы переходим в /home/bi/, там создается файл. Значит путь: /home/bi/имя файла



3.2 Скрипты на bash: основы 10 из 10 шагов пройдено 6 из 6 баллов получено

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [script1.sh](#), [script2.sh](#).

Предположим, что вы находитесь в директории `/home/bi/Documents/` и запускаете в ней скрипт следующего содержания:

```
#!/bin/bash
cd /home/bi/
touch file1.txt
cd /home/bi/Desktop/
```

Как будет выглядеть **абсолютный путь** до созданного файла `file1.txt` по окончанию работы скрипта?

Выберите один вариант из списка

Хорошая работа.

Верно решили 29 905 учащихся

Из всех попыток 76% верных

- /home/bi/Documents/file1.txt
- /home/bi/Desktop/file1.txt
- /home/bi/file1.txt
- Никак (файла file1.txt не будет существовать после завершения работы скрипта)

**Следующий шаг**

**Решить снова**

**Ваши решения** Вы получили: **1 балл**

Рис. 3.7: 3.2 Шаг 5

Шаг 7 - (рис. 3.8). В именах можно использовать буквы, нижние подчеркивания и цифры.

3.2 Скрипты на bash: основы 10 из 10 шагов пройдено 6 из 6 баллов получено

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [variables1.sh](#), [variables2.sh](#).

Какие из представленных ниже строк **могут** быть именами переменных в bash? Выберите **все** подходящие варианты!

**Подсказка:** если все варианты ответов являются неверными, то не отмечайте ни один из них и нажимайте кнопку "Отправить"/"Submit".

**Выберите все подходящие ответы из списка**

Отличное решение!

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

variable\_123  
 \_\_variable  
 var-i-able  
 var@iable  
 vari/able  
 variab\$le  
 variable123

**Следующий шаг**    **Решить снова**

Ваши решения Вы получили: **1 балл**

Верно решили **27 188** учащихся  
Из всех попыток **25%** верных

Рис. 3.8: 3.2 Шаг 7

Шаг 10 - (рис. 3.9). Здесь виден и ответ.

The screenshot shows a user interface for an online course. At the top, there is a navigation bar with icons for search, refresh, and other functions, along with a user icon labeled 'юп' with a green notification badge. Below the navigation bar, the course title '3.2 Скрипты на bash: основы' is displayed, along with progress information: '10 из 10 шагов пройдено' and '6 из 6 баллов получено'. A horizontal line separates this from the main content area.

**Вы можете скачать и изучить скрипт, который мы показали в видеофрагменте: [arguments.sh](#).**

Напишите скрипт на bash, который принимает на вход два аргумента и выводит на экран строку следующего вида:

```
Arguments are: $1=первый_аргумент $2=второй_аргумент
```

Например, если ваш скрипт называется `./script.sh`, то при запуске его `./script.sh one two` на экране должно появиться:

```
Arguments are: $1=one $2=two
```

а при запуске `./script.sh three four` будет:

```
Arguments are: $1=three $2=four
```

**Подсказка:** в случае проблем с решением задачи, обратите внимание [на наши рекомендации по написанию скриптов](#).

**Напишите программу. Тестируется через stdin → stdout**

Хорошая работа.

Верно решили 25 053 учащихся  
Из всех попыток 41% верных

Теперь вам доступен [Форум решений](#), где вы можете сравнить свое решение с другими или спросить совета.

```
1 echo "Arguments are: \$1=\$1 \$2=\$2"
2
3 echo "Arguments are: \$1=\$1 \$2=\$2"
```

Рис. 3.9: 3.2 Шаг 10, формулировка задания

Шаг 10, командный файл - (рис. 3.10).

The screenshot shows a terminal window titled 'yapolyakova1' with a status bar indicating '0 ● 1 gedit'. The window title is 'vncourse1.sh'. The file content is displayed in a code editor:

```
1 #!/bin/bash
2
3 echo "Arguments are: \$1=\$1 \$2=\$2"
```

Рис. 3.10: 3.2 Шаг 10, листинг

Шаг 10 - (рис. 3.11). Это проверка работы файла. Делаем его исполняемым и запускаем.

```

yapolyakova1@yapolyakova1:~$ chmod +x vncourse1.sh
yapolyakova1@yapolyakova1:~$ ./vncourse1.sh one two
Arguments are: $1=one $2=two
yapolyakova1@yapolyakova1:~$ █

```

Рис. 3.11: 3.2 Шаг 10

3. Часть “Скрипты на bash: ветвления и циклы”. Здесь 5 заданий.

Шаг 3 - (рис. 3.12). Здесь подошли все варианты.

3.3 Скрипты на bash: ветвления и циклы 9 из 9 шагов пройдено 10 из 10 баллов получено

Вы можете скачать и изучить скрипт, который мы показали в видеофрагменте: [branching1.sh](#).

Предположим, вы пишете скрипт на bash и хотите использовать в нем конструкцию `if` в следующем фрагменте:

```

if [[ ... ]]
then
    echo "True"
fi

```

Вы можете вписывать вместо “...” (внутри `[[ ... ]]` и не забудьте про пробелы после `[[` и перед `]]`) любое из перечисленных ниже условий. Однако мы просим вас выбрать только те из них, при которых `echo` напечатает на экран `True` вне зависимости от того, с какими параметрами был запущен ваш скрипт и какие в нем есть переменные.

Например, условие `0 -eq 0` **подходит**, т.к. ноль всегда равен нулю вне зависимости от аргументов и переменных внутри скрипта и на экран будет напечатано `True`. В то же время условие `$var1 -eq 0` **не подходит**, так как в переменной `var1` как может быть записан ноль (тогда будет напечатано `True`), так его может и не быть (тогда ничего напечатано не будет).

**Примечание:** если вы планируете проверять варианты ответов у себя в терминале, обратите внимание на то, что содержащие символ `$` тексты могут изменяться при копировании – не забудьте отредактировать их в соответствии с изображением на экране. Это связано с особенностями написания `$` в некоторых видах заданий на Stepik.

Выберите все подходящие ответы из списка

Хорошая работа.

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на них вопросы, или сравнить своё решение с другими на [форуме решений](#).

Верно решили 23 158 учащихся  
Из всех попыток 16% верных

- s \$0
- z ""
- \$var1 == \$var2 || \$var1 != \$var2
- !(4 <= 3)
- 5 -ge 5
- e \$0

Рис. 3.12: 3.3 Шаг 3

Шаг 5 - (рис. 3.13). 3 и 5 не проходят ни по одному из условий и идет в `else`



Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [branching2.sh](#), [branching3.sh](#).

Посмотрите на фрагмент bash-скрипта:

```
if [[ $var -gt 5 ]]
then
    echo "one"
elif [[ $var -lt 3 ]]
then
    echo "two"
elif [[ $var -eq 4 ]]
then
    echo "three"
else
    echo "four"
fi
```

Какие строки и в какой последовательности он выведет на экран, если сначала этот скрипт запустили задав переменную `var=3`, а затем запустили еще раз, но уже с `var=5`.

Выберите один вариант из списка

Абсолютно точно.

Верно решили 25 138 учащихся  
Из всех попыток 64% верных

- Сначала two, потом four
- Сначала four, потом four
- Сначала four, потом one
- Сначала two, потом one

Рис. 3.13: 3.3 Шаг 5

Шаг 6 - (рис. 3.14). Здесь формулировка задания



3.3 Скрипты на bash: ветвления и циклы 9 из 9 шагов пройдено 10 из 10 баллов получено

Напишите скрипт на bash, который принимает на вход один аргумент (целое число от 0 до бесконечности), который будет обозначать число студентов в аудитории. В зависимости от значения числа нужно вывести разные сообщения.

Соответствие входа и выхода должно быть таким:

```
0 --> No students
1 --> 1 student
2 --> 2 students
3 --> 3 students
4 --> 4 students
5 и больше --> A lot of students
```

Примечание а): выводить нужно только строку справа, т.е. "->" выводить не нужно.

Примечание б): в последней строке слово "lot" с маленькой буквы!

Примечание 2: в этой и всех последующих задачах на написание скриптов, если не указано явно, что нужно проверять вход (например, что он будет именно числом и именно от 0 до бесконечности), то этого делать не нужно!

Пример №1: если ваш скрипт называется `./script.sh`, то при запуске его как `./script.sh 1` на экране должно появиться:

```
1 student
```

Пример №2: если ваш скрипт называется `./script.sh`, то при запуске его как `./script.sh 5` на экране должно появиться:

```
A lot of students
```

Подсказка: в случае проблем с решением задачи, обратите внимание [на наши рекомендации по написанию скриптов](#).

Напишите программу. Тестируется через stdin → stdout

Верно.

Верно решили 23 310 учащихся  
Из всех попыток 38% верных

Рис. 3.14: 3.3 Шаг 6, формулировка задания

Шаг 6 - (рис. 3.15). Командный файл, который решает задачу.

```
1 #!/bin/bash
2
3 students = $1
4
5 case "$students" in
6     0) echo "No students" ;;
7     1) echo "1 student" ;;
8     2) echo "2 students" ;;
9     3) echo "3 students" ;;
10    4) echo "4 students" ;;
11    *) echo "A lot of students" ;;
12 esac|
```

Рис. 3.15: 3.3 Шаг 6, командный файл

Шаг 6 - (рис. 3.16). Делаем файл исполняемым, проверяем с разными данными

```
yapolyakova1@yapolyakova1:~$ chmod +x vncourse2.sh
yapolyakova1@yapolyakova1:~$ ./vncourse2.sh 0
No students
yapolyakova1@yapolyakova1:~$ ./vncourse2.sh 1
1 student
yapolyakova1@yapolyakova1:~$ ./vncourse2.sh 2
2 students
yapolyakova1@yapolyakova1:~$ ./vncourse2.sh 3
3 students
yapolyakova1@yapolyakova1:~$ ./vncourse2.sh 4
4 students
yapolyakova1@yapolyakova1:~$ ./vncourse2.sh 5
A lot of students
yapolyakova1@yapolyakova1:~$ █
```

Рис. 3.16: 3.3 Шаг 6, проверка работы

Шаг 8 - (рис. 3.17). Ответ 5 раз “старт” и 4 раза “финиш”. Я проверила это создав и запустив командный файл.



3.3 Скрипты на bash: ветвления и циклы 9 из 9 шагов пройдено 10 из 10 баллов получено

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [loops1.sh](#), [loops2.sh](#).

Посмотрите на фрагмент bash-скрипта:

```
for str in a , b , c_d
do
    echo "start"
    if [[ $str > "c" ]]
    then
        continue
    fi
    echo "finish"
done
```

Если запустить этот скрипт, то сколько раз на экран будет выведено слово "start", а сколько раз слово "finish"?

Выберите один вариант из списка

Верно решили 24 582 учащихся  
Из всех попыток 45% верных

- 5 раз "start" и 2 раза "finish"
- 3 раза "start" и ни разу "finish"
- 5 раз "start" и ни разу "finish"
- 5 раз "start" и 4 раза "finish"

Отправить

[Ваши решения](#) Вы получили: 1 балл

Рис. 3.17: 3.3 Шаг 8

Шаг 9 - (рис. 3.18). Здесь задание



3.3 Скрипты на bash: ветвления и циклы 9 из 9 шагов пройдено 10 из 10 баллов получено

Напишите скрипт на bash, который будет определять в какую возрастную группу попадают пользователи. При запуске скрипта должен вывести сообщение "enter your name:" и ждать от пользователя ввода имени (используйте `read`, чтобы прочитать его). Когда имя введено, то скрипт должен написать "enter your age:" и ждать ввода возраста (опять нужен `read`). Когда возраст введен, скрипт пишет на экран "<Имя>, your group is <группа>", где <группа> определяется на основе возраста по следующим правилам:

- младше либо равно 16: "child",
- от 17 до 25 (включительно): "youth",
- старше 25: "adult".

После этого скрипт опять выводит сообщение "enter your name:" и всё начинается по новой (бесконечный цикл!). Если в какой-то момент работы скрипта будет введено **пустое имя или возраст 0**, то скрипт должен написать на экран "bye" и закончить свою работу (выход из цикла!).

Примеры корректной работы скрипта:

№1

```
./script.sh
enter your name:
Egor
enter your age:
16
Egor, your group is child
enter your name:
Elena
enter your age:
0
bye
```

№2:

```
./script.sh
enter your name:
```

Рис. 3.18: 3.3 Шаг 9, формулировка задания

Шаг 9 - (рис. 3.19). Командный файл-ответ

The screenshot shows a terminal window titled 'yapolyakova1' with a sub-titler bar '0 ● 1 bash'. The main area displays a shell script named 'vncourse2.sh'. The script content is as follows:

```
1 #!/bin/bash
2
3 # Начинаем бесконечный цикл
4 while true; do
5     # Просим ввести имя
6     echo "enter your name:"
7     read user_name
8
9     # Выход, если имя пустое
10    if [[ -z "$user_name" ]]; then
11        echo "bye"
12        exit 0
13    fi
14    # Просим ввести возраст
15    echo "enter your age:"
16    read user_age
17    # Проверяем возраст на значение '0' и выполняем выход
18    if [[ "$user_age" -eq 0 ]]; then
19        echo "bye"
20        exit 0
21    fi
22    # Определяем возрастную группу
23    if [[ "$user_age" -le 16 ]]; then
24        group="child"
25    elif [[ "$user_age" -ge 17 && "$user_age" -le 25 ]]; then
26        group="youth"
27    else
28        group="adult"
29    fi
30    # Показываем пользователю его возрастную группу
31    echo "$user_name, your group is $group"
32 done
```

Рис. 3.19: 3.3 Шаг 9, командный файл

Шаг 9 - (рис. 3.20). Открываем доступ на исполнение, запускаем с разными данными.

```
yapolyakova1@yapolyakova1:~$ chmod +x vncourse3.sh
yapolyakova1@yapolyakova1:~$ ./vncourse3.sh
enter your name:
Julia
enter your age:
18
Julia, your group is youth
enter your name:
masha
enter your age:
3
masha, your group is child
enter your name:
misha
enter your age:
50
misha, your group is adult
enter your name:
0
enter your age:
0
bye
yapolyakova1@yapolyakova1:~$
```

Рис. 3.20: 3.3 Шаг 9, проверка работы

4. Часть “Скрипты на bash: разное”. Здесь 6 заданий.

Шаг 3 - (рис. 3.21). В первом неподходящем варианте + и = не в том порядке, во втором - нет let, остальные подходят.



3.4 Скрипты на bash: разное 10 из 10 шагов пройдено 14 из 14 баллов получено

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [math1.sh](#), [math2.sh](#).

Какие(ая) из предложенных ниже инструкций увеличает значение переменной `a` на значение переменной `b`? Например, если в `a` было записано 10, в `b` было 5, то в `a` должно записаться 15.

Выберите **все подходящие** варианты!

**Примечание:** если вы планируете проверять варианты ответов у себя в терминале, обратите внимание на то, что содержащие символ `$` тексты могут изменяться при копировании – не забудьте отредактировать их в соответствии с изображением на экране. Это связано с особенностями написания `$` в некоторых видах заданий на Stepik.

**Подсказка:** обратите особое внимание на кавычки и пробелы, они могут как принципиально изменить команду, так и ни на что не повлиять (в зависимости от команды и контекста)!

Выберите **все подходящие** ответы из списка

Хорошая работа.

Верно решили 22 116 учащихся  
Из всех попыток 20% верных

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

- let a=a+b
- let "a+=b"
- a+=\$b
- let "a+=b"
- let a=\$a+\$b

Следующий шаг

Решить снова

Рис. 3.21: 3.4 Шаг 3

Шаг 5 - (рис. 3.22). Мы перешли командой `cd` в директорию, значит `pwd` выведет именно ее путь.

The screenshot shows a user interface for an online programming exercise. At the top, there's a navigation bar with icons for back, forward, and search, along with a '0' icon and a 'юп' button. Below the bar, a progress message reads: '3.4 Скрипты на bash: разное 10 из 10 шагов пройдено 14 из 14 баллов получено'. The main content area contains a question: 'Вы можете скачать и изучить скрипт, который мы показали в видеофрагменте: [programs.sh](#). Пусть вы находитесь в директории /home/bi/Documents/ и запускаете в ней скрипт следующего содержания:' followed by a code block:

```
#!/bin/bash
cd /home/bi/
echo "`pwd`"
```

Что в этом случае выведет команда echo на экран?

Выберите один вариант из списка

Отличное решение!

Верно решили 23 677 учащихся  
Из всех попыток 51% верных

/home/bi/Documents  
 /home/bi  
 pwd  
 `pwd'  
 Код возврата команды pwd (0 в случае успешного выполнения и не 0 в случае ошибок)

**Следующий шаг**    **Решить снова**

Ваши решения Вы получили: 1 балл

Рис. 3.22: 3.4 Шаг 5

Шаг 6 - (рис. 3.23). Программа сначала должна вывести результат на экран или в файл, а потом мы сможем работать с кодом возврата.



3.4 Скрипты на bash: разное 10 из 10 шагов пройдено 14 из 14 баллов получено

Мы рассказали, что можно проверить код возврата внешней программы прямо в конструкции `if` при помощи `if `program options arguments`` (действия внутри `if` выполняются, если программа закончилась с кодом 0). Однако это не всегда правда! Если запуск внешней программы выводит что-то в `stdout`, то в проверку `if` поступит именно этот вывод, а не код возврата! Вы можете убедиться в этом, написав простой bash-скрипт с использованием, например, `if `pwd``.

Однако как быть, если хочется всё-таки запустить программу `program`, которая пишет что-то в `stdout` и потом выполнить какие-то действия если ее код возврата равен 0? Выберите **все верные** утверждения или правильно работающие конструкции `if`.

**Примечание:** во всех вариантах ответов, где есть кавычка, используется именно **косая кавычка** (`), а не обычная (') или двойная (").

Выберите все подходящие ответы из списка

Всё правильно.

Верно решили 21 426 учащихся

Из всех попыток 20% верных

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

- `if [[ `program` -eq 0 ]]`
- `if 'program > some_file.txt'`
- Ничего сделать нельзя
- Сначала запустить `program`, затем `if [[ $? -eq 0 ]]`
- Сначала `var='program'`, затем `if [[ $var -eq 0 ]]`

[Следующий шаг](#)

[Решить снова](#)

Ваши решения Вы получили: 1 балл

Рис. 3.23: 3.4 Шаг 6

Шаг 8 - (рис. 3.24). Эту задачу я проверила у себя, затем скопировала строку, которая выводится.



3.4 Скрипты на bash: разное 10 из 10 шагов пройдено 14 из 14 баллов получено

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [functions1.sh](#), [functions2.sh](#).

Посмотрите на функцию из bash-скрипта:

```
counter () # takes one argument
{
    local let "c1+=$1"
    let "c2+=$((1}*2"
}
```

Впишите в форму ниже строку, которую выведет на экран команда `echo "counters are $c1 and $c2"` если она находится в скрипте **после десяти вызовов** функции `counter` с параметрами сначала 1, затем 2, затем 3 и т.д., последний вызов с параметром 10.

**Подсказка:** этот пример можно решить в уме, но если система проверки не принимает ваше решение, то возможно вы что-то упустили (возможно что-то совсем небольшое/невидимое 😊). В этом случае имеет смысл написать небольшой скрипт на bash, который проделает ровно то, что указано в задании и посимвольно сверить свой ответ с тем, что он выдаст на экран.

Напишите текст

✓ Хорошие новости, верно!

Верно решили 20 009 учащихся  
Из всех попыток 28% верных

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

counters are 1 and 110

Рис. 3.24: 3.4 Шаг 8

Шаг 9 - (рис. 3.25). Формулировка задания.

3.4 Скрипты на bash: разное 14 из 10 шагов пройдено 14 из 14 баллов получено

Напишите скрипт на bash, который будет искать наибольший общий делитель (НОД, greatest common divisor, GCD) двух чисел. При запуске ваш скрипт не должен ничего писать на экран, а просто ждет ввода двух натуральных чисел через пробел (для этого можно использовать `read` и указать ему две переменные – см. пример в видеофрагменте). После ввода чисел скрипт считает их НОД и выводит на экран сообщение "GCD is <посчитанное значение>", например, для чисел 15 и 25 это будет "GCD is 5". После этого скрипт опять входит в режим ожидания двух натуральных чисел. Если в какой-то момент работы пользователь ввел вместо этого пустую строку, то нужно написать на экран "bye" и закончить свою работу.

Вычисление НОД несложно реализовать с помощью [алгоритма Евклида](#). Вам нужно написать функцию `gcd`, которая принимает на вход два аргумента (назовем их **M** и **N**). **Если аргументы равны**, то мы нашли НОД – он равен **M** (или **N**), нужно выводить соответствующее сообщение на экран (см. выше). Иначе нужно сравнить аргументы между собой. Если **M больше N**, то запускаем ту же функцию `gcd`, но в качестве первого аргумента передаем (**M-N**), а в качестве второго **N**. Если же наоборот, **M меньше N**, то запускаем функцию `gcd` с первым аргументом **M**, а вторым (**N-M**).

Пример корректной работы скрипта:

```
./script.sh  
10 15  
GCD is 5  
7 3  
GCD is 1  
bye
```

**Примечание:** в вызове функции из себя самой нет ничего страшного или неправильного, т.ч. смело вызывайте `gcd` прямо внутри `gcd`!

**Примечание 2:** для завершения работы функции в произвольном месте, можно использовать инструкцию `return` (все инструкции функции после `return` выполняться не будут). В отличии от `exit` эта команда завершит только функцию, а не выполнение всего скрипта целиком. Однако в данной задаче можно обойтись и без использования `return`!

**Подсказка:** в случае проблем с решением задачи, обратите внимание [на наши рекомендации по написанию скриптов](#).

Рис. 3.25: 3.4 Шаг 9, формулировка

Шаг 9 - (рис. 3.26). Листинг командного файла.

The screenshot shows a terminal window titled 'yapolyakova1 - Thunar' with a sub-titler bar 'yapolyakova1 □ 0 ● 1 bash'. Below the title bar is a toolbar with 'Open' and a dropdown menu. The main area is a code editor showing a bash script named 'vncourse5.sh'. The script contains the following code:

```
1 #!/bin/bash
2
3 while [ true ]
4 do
5     read n1 n2
6     if [ -z $n1 ]; then
7         echo "bye"
8         break
9     else
10        gcd () {
11            remainder=1
12            if [ $n2 -eq 0 ]
13            then
14                echo "bye"
15                fi
16            while [ $remainder -ne 0 ]
17            do
18                remainder=$((n1%n2))
19                n1=$n2
20                n2=$remainder
21            done
22        }
23        gcd $1 $2
24        echo "GCD is $n1"
25    fi
26 done
```

Рис. 3.26: 3.4 Шаг 9, командный файл

Шаг 9 - (рис. 3.27). Как обычно, открываем доступ на исполнение, запускаем, тестируем.

The screenshot shows a terminal window with the following session:

```
yapolyakova1@yapolyakova1:~$ chmod +x vncourse5.sh
yapolyakova1@yapolyakova1:~$ ./vncourse5.sh
10 15
GCD is 5
7 3
GCD is 1

bye
yapolyakova1@yapolyakova1:~$
```

Рис. 3.27: 3.4 Шаг 9, проверка работы

Шаг 10 - (рис. 3.28). Формулировка задания.



Напишите **калькулятор** на bash. При запуске ваш скрипт должен ожидать ввода пользователем команды (при этом на экран выводить ничего не нужно). Команды могут быть трех типов:

1. Слово "exit". В этом случае скрипт должен вывести на экран слово "bye" и завершить работу.
2. Три аргумента через пробел – первый operand (целое число), операция (одна из "+", "-", "\*", "/", "%", "\*\*") и второй operand (целое число). В этом случае нужно произвести указанную операцию над заданными числами и вывести результат на экран. После этого переходим в режим ожидания новой команды.
3. Любая другая команда из одного аргумента или из трех аргументов, но с операцией не из списка. В этом случае нужно вывести на экран слово "error" и завершить работу.

Чтобы проверить работу скрипта, вы можете записать сразу несколько команд в файл и передать его скрипту на stdin (т.е. выполнить `./script.sh < input.txt`). В этом случае он должен вывести сразу все ответы на экран.

Например, если входной файл будет следующего содержания:

```
10 + 1
2 ** 10
exit
```

то на экране будет:

```
11
1024
bye
```

Если же на вход поступит следующий файл:

```
3 - 5
2/10
exit
```

то на экране будет:

```
-2
error
```

Рис. 3.28: 3.4 Шаг 10, формулировка

Шаг 10 - (рис. 3.29). Листинг командного файла калькулятора.

The screenshot shows a terminal window titled 'yapolyakova1 - Thunar' with a sub-titler bar '0 ● 1 bash'. The window contains a file named 'kalkulator.sh' with the following content:

```
1 #!/bin/bash
2
3 while [[ True ]]
4 do
5     read first operator second
6     if [[ $first == "exit" ]]
7     then
8         echo "bye"
9         break
10    elif [[ "$first" =~ ^[0-9]+\$" && "$second" =~ ^[0-9]+\$" ]]
11    then
12        echo "error"
13        break
14    else
15        case $operator in
16        "+") let "result = first + second";;
17        "-") let "result = first - second";;
18        "/") let "result = first / second";;
19        "*") let "result = first * second";;
20        "%") let "result = first % second";;
21        "**") let "result = first ** second";;
22    *) echo "error" ; break ;;
23    esac
24    echo "$result"
25 fi
26 done
```

Рис. 3.29: 3.4 Шаг 10, командный файл

Шаг 10 - (рис. 3.30). Как обычно, открываем доступ на исполнение, запускаем, тестируем.

```

yapolyakova1@yapolyakova1:~$ chmod +x kalkulator.sh
yapolyakova1@yapolyakova1:~$ ./kalkulator.sh
10 + 1
11
2 ** 10
1024
exit
bye
yapolyakova1@yapolyakova1:~$ ./kalkulator.sh
3 - 5
-2
2/10
error
yapolyakova1@yapolyakova1:~$ 

```

Рис. 3.30: 3.4 Шаг 10, проверка работы

5. Часть “Продвинутый поиск и редактирование”. Здесь 7 заданий.

Шаг 3 - (рис. 3.31). Первой команде не важен регистр, а второй важен (из-за опций `iname` и `name`)

3.5 Продвинутый поиск и редактирование 13 из 13 шагов пройдено 10 из 10 баллов получено

Пусть в директории `/home/bi` лежат файлы `Star_Wars.avi`, `star_trek_OST.mp3`, `STARS.txt`, `stardust.mpeg`, `Eddard_Stark_biography.txt`.

Отметьте все файлы, которые **найдет** команда `find /home/bi -iname "star*"`, но **НЕ найдет** команда `find /home/bi -name "star*"`?

**Выберите все подходящие ответы из списка**

Eddard\_Stark\_biography.txt  
 star\_trek\_OST.mp3  
 stardust.mpeg  
 STARS.txt  
 Star\_Wars.avi

**Верно решили 20 547 учащихся**  
**Из всех попыток 36% верных**

Следующий шаг    Решить снова

Ваши решения Вы получили: 1 балл

Рис. 3.31: 3.5 Шаг 3

Шаг 4 - (рис. 3.32). Здесь можно понять, рассуждая логически, основываясь на материалах теории из видео.

The screenshot shows a user interface for an online learning platform. At the top, there is a navigation bar with icons for back, forward, and search, along with a user icon and a counter '0'. Below the navigation bar, a progress bar indicates '3.5 Продвинутый поиск и редактирование' has been completed, showing '13 из 13 шагов пройдено' and '10 из 10 баллов получено'. A message below the progress bar reads: 'Задание на понимание работы опций `-path` и `-name` команды `find`. Отметьте **все верные** утверждения из перечисленных ниже.' In the main area, a section titled 'Выберите все подходящие ответы из списка' contains several statements with checkboxes. Some checkboxes are checked (green checkmark) and others are empty (grey outline). A message box says: 'Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#)'. Below the list of statements, two buttons are visible: 'Следующий шаг' (Next step) and 'Решить снова' (Solve again). At the bottom, a message says: 'Ваши решения Вы получили: 1 балл'.

Рис. 3.32: 3.5 Шаг 4

Шаг 5 - (рис. 3.33). Глубина `/home/bi = 1`, `dir1 = 2`, `dir2 = 3`. Поэтому будет только первый и второй файлы.

3.5 Продвинутый поиск и редактирование 13 из 13 шагов пройдено 10 из 10 баллов получено

Предположим, что в директории `/home/bi/` есть следующая структура файлов и поддиректорий:

```
/home/bi/
└── dir1
    ├── file1
    └── dir2
        ├── file2
        └── dir3
            └── file3
```

Какие(ой) из трех файлов (`file1`, `file2`, `file3`) будут найдены по команде `find /home/bi -mindepth 2 -maxdepth 3 -name "file*"`?

Выберите один вариант из списка

Отлично!

Верно решили 20 711 учащихся  
Из всех попыток 41% верных

Все три файла  
 Все кроме `file2`  
 Только `file1`  
 Все кроме `file1`  
 Все кроме `file3`

Следующий шаг      Решить снова

Ваши решения Вы получили: 1 балл

Рис. 3.33: 3.5 Шаг 5

Шаг 7 - (рис. 3.34). А - num строк конечного контекста после сопоставления, В - num строк начального контекста перед соотв. строками, С - num строк начального и конечного выходного контекста.

3.5 Продвинутый поиск и редактирование 13 из 13 шагов пройдено 10 из 10 баллов получено

Задание на понимание работы опций `-A`, `-B` и `-C` команды `grep`. Пусть у вас есть файл `file.txt` из 10 строк, причем **в каждой строке есть слово "word"**. Если вы выполните на этом файле команды:

```
grep "word" file.txt > results.txt
grep -A 1 "word" file.txt > results.txt
grep -B 1 "word" file.txt > results.txt
grep -C 1 "word" file.txt > results.txt
```

то какая(ие) из них создаст файл `results.txt` наибольшего размера?

Выберите один вариант из списка

Так точно!

Верно решили 20 237 учащихся  
Из всех попыток 41% верных

Все, кроме `grep "word" file.txt > results.txt`  
 `grep -A 1 "word" file.txt > results.txt` и `grep -B 1 "word" file.txt > results.txt`  
 `grep -A 1 "word" file.txt > results.txt`  
 `results.txt` будет одинакового размера во всех случаях  
 `grep -C 1 "word" file.txt > results.txt`

Следующий шаг Решить снова

Ваши решения Вы получили: 1 балл

Рис. 3.34: 3.5 Шаг 7

Шаг 9 - (рис. 3.35). Первая буква в Ubuntu может быть большой или маленькой, перед словом должно быть любое сочетание по порядку из скобок левее. В неправильном варианте нет и

3.5 Продвинутый поиск и редактирование 13 из 13 шагов пройдено 10 из 10 баллов получено

Предположим, что в файле `text.txt` записаны строки, показанные среди вариантов ответа. Отметьте только те из них, которые выведет на экран команда `grep -E "[xkLXKL]?[uU]buntu$" text.txt`.

**Выберите все подходящие ответы из списка**

Хорошие новости, верно!

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

The best OS is Xubuntu  
 Linux is not always Ubuntu  
 Lubuntu is better than Ubuntu  
 Kubuntu  
 Hmm, XKLubuntu  
 I prefer Kubuntu

**Верно решили 18 768 учащихся**  
Из всех попыток 23% верных

**Следующий шаг**    **Решить снова**

Ваши решения Вы получили: **2 балла**

Рис. 3.35: 3.5 Шаг 9

Шаг 11 - (рис. 3.36). -n ключ подавляет автоматический вывод шаблона

3.5 Продвинутый поиск и редактирование 13 из 13 шагов пройдено 10 из 10 баллов получено

Что произойдет, если в команде `sed -n "/[a-z]*p" text.txt` не указывать опцию `-n`?

**Выберите один вариант из списка**

Всё правильно.

**Верно решили 19 784 учащихся**  
Из всех попыток 39% верных

На экран ничего не напечатается  
 Каждая строчка будет выведена два раза  
 На экран будет выведено всё содержимое файла `text.txt`  
 Будут выведены все строки файла `text.txt`, в которых есть только большие буквы латинского алфавита

**Следующий шаг**    **Решить снова**

Ваши решения Вы получили: **1 балл**

Рис. 3.36: 3.5 Шаг 11

Шаг 12 - (рис. 3.37). В `sed` передается регулярное выражение замены `s`.



3.5 Продвинутый поиск и редактирование 13 из 13 шагов пройдено 10 из 10 баллов получено

Запишите в форму ниже инструкцию `sed`, которая заменит все "аббревиатуры" в файле `input.txt` на слово "abbreviation" и запишет результат в файл `edited.txt` (на экран при этом ничего выводить не нужно). Обратите внимание, что **в инструкции должны быть указаны и сам `sed`, и оба файла!**

Под "аббревиатурой" будем понимать слово, которое удовлетворяет следующим условиям:

- состоит только из больших букв латинского алфавита,
- состоит из хотя бы двух букв,
- окружено одним пробелом с каждой стороны.

При этом будем считать, что в тексте **не может быть две "аббревиатуры" подряд**. Например, текст " YOU YOU and YOU!" является **некорректным** (в нем есть две "аббревиатуры", но они идут подряд) и на таких примерах мы проверять вашу инструкцию **не будем**.

**Пример:** если у вас был текст "Hi, I heard these songs by ABBA, TLA and DM!", то он должен быть преобразован в "Hi, I heard these songs by ABBA, abbreviation and abbreviation!" .

**Примечание:** после вашей замены "аббревиатуры" на слово "abbreviation" **количество пробелов** в тексте **не должно меняться!**

**Внимание!** Во время проверки **мы не запускаем команду**, которую вы ввели на реальном файле "аббревиатурами" (это небезопасно, можно же ввести `rm -rf /*`!). Вместо этого мы сперва анализируем структуру вашей инструкции (например, что в ней использован именно `sed` и сделано это ровно один раз, что на вход подается `input.txt`, а результат будет записан в `edited.txt` и т.д.), а затем **запускаем её смысловую часть** (т.е. поиск по регулярному выражению и замена на "abbreviation") на тестовых примерах. К сожалению, наш запуск не идеально повторяет `sed`, но он очень близок к нему. Главная "несовместимость" заключается в том, что наша проверка не понимает идущие подряд символы, отвечающие за количество повторений (т.е. \*, +, ? и {}). Однако эту "несовместимость" легко исправить указав при помощи "(" и ")" какой из символов к чему относится! Например, регулярное выражение `a+?` (ноль или один раз по одной или более букве "a") нужно записать как `(a+)?` (при этом запись `(a)+?`, конечно же, не поможет).

#### Напишите текст

Отличное решение!

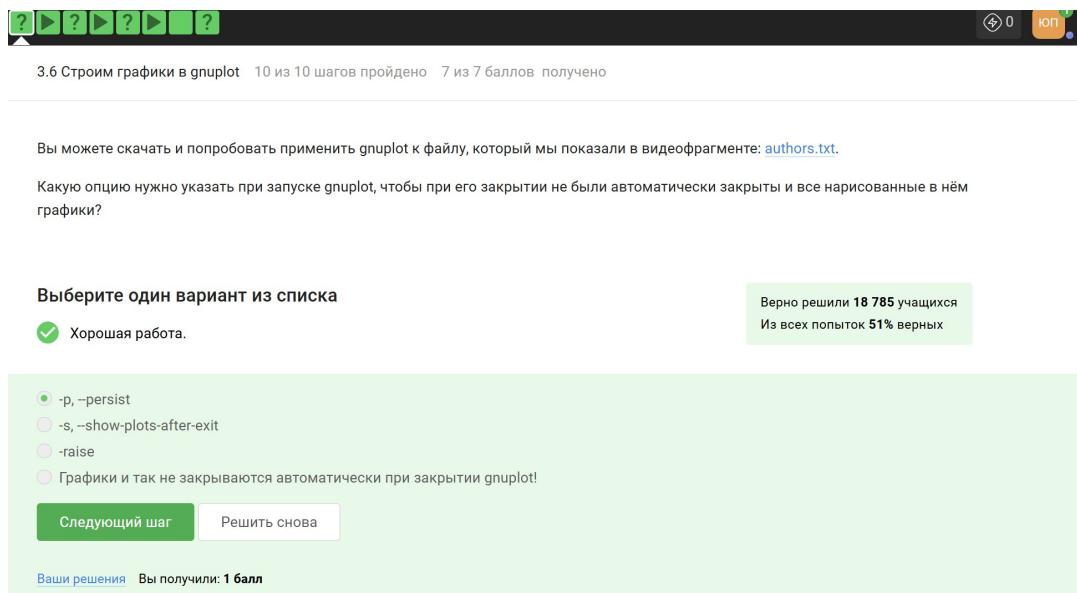
Верно решили **16 632** учащихся  
Из всех попыток **34%** верных

`sed 's/ [A-Z]\{2,\} / abbreviation /g' input.txt >edited.txt'`

Рис. 3.37: 3.5 Шаг 12

#### 6. Часть "Строим графики в gnuplot". Здесь 4 задания.

Шаг 3 - (рис. 3.38). Эта опция была показана в видео.



Вы можете скачать и попробовать применить gnuplot к файлу, который мы показали в видеофрагменте: [authors.txt](#).

Какую опцию нужно указать при запуске gnuplot, чтобы при его закрытии не были автоматически закрыты и все нарисованные в нём графики?

**Выберите один вариант из списка**

-p, -persist  
 -s, --show-plots-after-exit  
 -raise  
 Графики и так не закрываются автоматически при закрытии gnuplot!

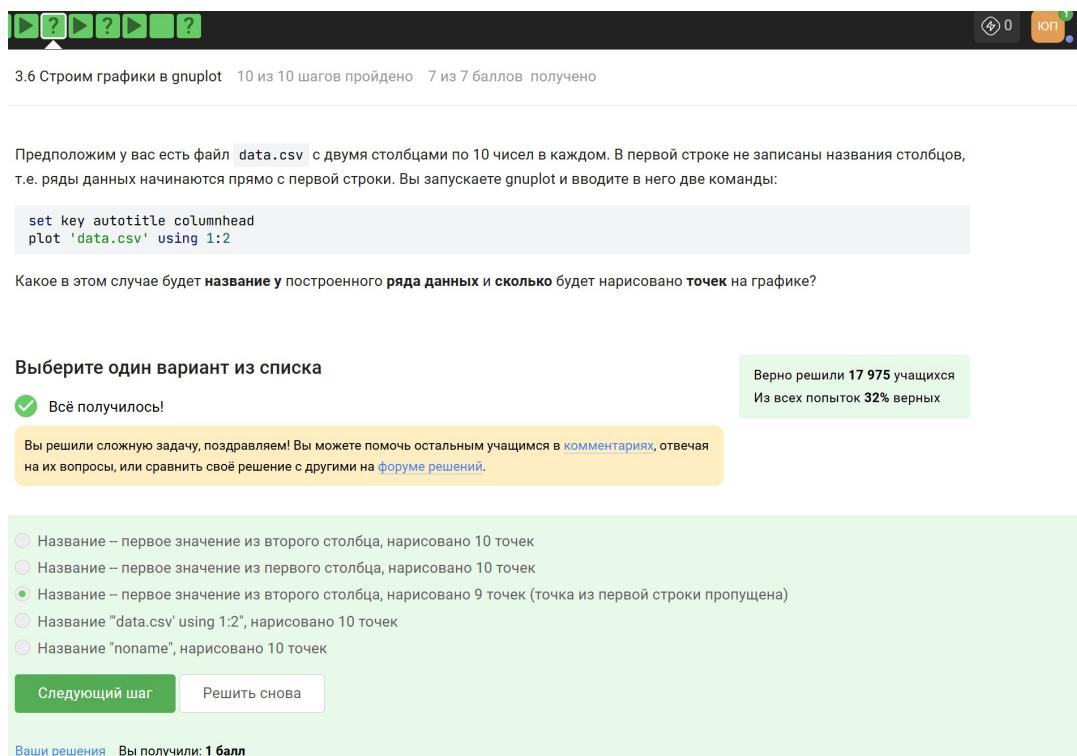
**Верно решили 18 785 учащихся**  
Из всех попыток 51% верных

**Следующий шаг**    **Решить снова**

Ваши решения Вы получили: **1 балл**

Рис. 3.38: 3.6 Шаг 3

Шаг 5 - (рис. 3.39). Ответ основан на материалах из видео теории.



Предположим у вас есть файл `data.csv` с двумя столбцами по 10 чисел в каждом. В первой строке не записаны названия столбцов, т.е. ряды данных начинаются прямо с первой строки. Вы запускаете gnuplot и вводите в него две команды:

```
set key autotitle columnhead
plot 'data.csv' using 1:2
```

Какое в этом случае будет **название** у построенного **ряда данных** и сколько будет нарисовано **точек** на графике?

**Выберите один вариант из списка**

Всё получилось!

Вы решили сложную задачу, поздравляем! Вы можете помочь остальным учащимся в [комментариях](#), отвечая на их вопросы, или сравнить своё решение с другими на [форуме решений](#).

**Верно решили 17 975 учащихся**  
Из всех попыток 32% верных

Название – первое значение из второго столбца, нарисовано 10 точек  
 Название – первое значение из первого столбца, нарисовано 10 точек  
 Название – первое значение из второго столбца, нарисовано 9 точек (точка из первой строки пропущена)  
 Название "data.csv" using 1:2", нарисовано 10 точек  
 Название "noname", нарисовано 10 точек

**Следующий шаг**    **Решить снова**

Ваши решения Вы получили: **1 балл**

Рис. 3.39: 3.6 Шаг 5

Шаг 7 - (рис. 3.40). Сначала идет команда установки подписей, а потом в скобках

## ПОДПИСЬ.

3.6 Строим графики в gnuplot 10 из 10 шагов пройдено 7 из 7 баллов получено

Вы можете скачать и изучить скрипты, которые мы показали в видеофрагменте: [plot.gnu](#), [plot\\_advanced.gnu](#), [plot\\_advanced2.gnu](#). Все три скрипта основаны на [этой заметке](#), данные также взяты оттуда.

Предположим, что вы пишите gnuplot-скрипт и у вас в нем есть три переменные `x1`, `x2`, `x3`, в которых записаны координаты важных точек по оси ОХ (по возрастанию). Вы хотите, чтобы на этой оси было только три деления (т.е. три черточки) в этих самых координатах, а подписи этих делений были оформлены в виде "point <номер точки>, value <значение соответствующей переменной>".

Например, для `x1=0, x2=10, x3=20`, это были бы надписи "point 1, value 0" в точке с координатой 0 по горизонтали, "point 2, value 10" в точке с координатой 10 и "point 3, value 20" в точке с координатой 20.

Или, например, `x1=100, x2=150, x3=250`, это были бы надписи "point 1, value 100" в точке с координатой 100, "point 2, value 150" в точке с координатой 150 и "point 3, value 250" в точке с координатой 250.

Впишите в форму ниже **одну команду** (т.е. одну строку), которую нужно добавить в скрипт, для выполнения этой задачи.

**Примечание:** проверять, что переменные `x1`, `x2`, `x3` идут по возрастанию или что они являются числами **не нужно**!

**Примечание 2:** в видеофрагменте на предыдущем шаге звучал термин **конкатенация**, который важен для выполнения данного задания. Под **конкатенацией** обычно понимают "склеивание" двух строк в одну длинную строку, например, конкатенация строк "Данные из файла" и "data.csv" даст строку "Данные из файла data.csv".

**Подсказка:** настоятельно рекомендуем изучить примеры скриптов – в них есть большая часть решения!

**Напишите текст**

Правильно, молодец!

Верно решили 13 935 учащихся  
Из всех попыток 44% верных

```
set xtics ("point 1, value ".x1 x1, "point 2, value ".x2 x2, "point 3, value ".x3 x3)
```

Рис. 3.40: 3.6 Шаг 7

Шаг 10 - (рис. 3.41). Формулировка задания.



3.6 Строим графики в gnuplot 10 из 10 шагов пройдено 7 из 7 баллов получено

Если вы не скачали на предыдущем шаге файлы [animated.gnu](#) и [move.rot](#), то скачайте их теперь, т.к. они понадобятся для выполнения задания.

Указанные файлы использовались в последнем видеофрагменте для создания вращающегося графика. Измените инструкции в файле `move.rot` (т.е. **добавлять и удалять инструкции нельзя!**) таким образом, чтобы:

- График **отразился зеркально** относительно горизонтальной поверхности. То есть там, где была точка (10, 10, 200), станет точка (10, 10, -200), где была точка (-10, -10, 200) станет (-10, -10, -200) и т.д. При этом точка (0, 0, 0) останется на месте.
- Изображение стало **вращаться в обратную сторону**. То есть если раньше вращалось "влево", то теперь станет "вправо".
- Вращение стало **в два раза быстрее**. То есть станет в два раза больше перерисовок графика на каждую секунду вращения.

Измененный файл загрузите в форму ниже.

**Примечание:** наша система проверки **не может** запустить на вашем файле `move.rot` программу gnuplot и сравнить полученный график с заданным. Вместо этого **мы анализируем команды**, которые вы указали в файле. Поэтому если вы видите, что ваш скрипт в gnuplot работает точно по условию, а мы отвечаем "Incorrect/Неверно", то попробуйте упростить свою модификацию `move.rot` и отправить его еще раз.

Напишите текст

Хорошая работа.

Верно решили 12 854 учащихся  
Из всех попыток 47% верных

`move.rot (94 bytes)`

**Следующий шаг**

**Решить снова**

Ваши решения Вы получили: 3 балла

Рис. 3.41: 3.6 Шаг 10, формулировка

Шаг 10 - (рис. 3.42). Меняем 10 на 350, ставим минусы у координат, уменьшаем паузу в 2 раза.

The screenshot shows a window titled "move.rot" with a menu bar in Russian: "Файл" (File), "Изменить" (Edit), and "Просмотр" (View). The main area contains the following text:

```
a=a+1
zrot=(zrot+350)%360
set view xrot,zrot
splot -x**2-y**2
pause 0.1
if (a<50) reread
```

Рис. 3.42: 3.6 Шаг 10, измененный файл

## **4 Выводы**

Был полностью завершен третий раздел внешнего курса.