

Выполнение внешнего курса. Блок №3. Продвинутые темы

Отчёт

Сергеев Даниил Олегович

Содержание

1	Цель работы	6
2	Задание	7
3	Выполнение лабораторной работы	8
3.1	Текстовый редактор vim	8
3.2	Скрипты на bash: основы	12
3.3	Скрипты на bash: ветвления и циклы	14
3.4	Скрипты на bash: разное	22
3.5	Продвинутый поиск и редактирование	32
3.6	Строим графики в gnuplot	36
3.7	Разное	39
4	Вывод	43
	Список литературы	44

Список иллюстраций

3.1	Задание №1. Условие и ответ	8
3.2	Задание №2. Условие и ответ	9
3.3	Задание №3. Условие и ответ	10
3.4	Задание №4. Условие и ответ	11
3.5	Задание №5. Условие и ответ	11
3.6	Задание №6. Условие и ответ	12
3.7	Задание №7. Условие и ответ	13
3.8	Задание №8. Условие и ответ	13
3.9	Задание №9. Условие и ответ	14
3.10	Задание №10. Условие и ответ	15
3.11	Задание №11. Условие и ответ	16
3.12	Задание №12. Условие	17
3.13	Задание №12. Ответ	17
3.14	Задание №13. Условие и ответ	19
3.15	Задание №14. Условие	20
3.16	Задание №14. Ответ	21
3.17	Задание №15. Условие и ответ	23
3.18	Задание №16. Условие и ответ	24
3.19	Задание №17. Условие и ответ	24
3.20	Задание №18. Условие и ответ	25
3.21	Задание №19. Условие	26
3.22	Задание №19. Ответ	27
3.23	Задание №20. Условие	29
3.24	Задание №20. Ответ	30
3.25	Задание №21. Условие и ответ	32
3.26	Задание №22. Условие и ответ	32
3.27	Задание №23. Условие и ответ	33
3.28	Задание №24. Условие и ответ	34
3.29	Задание №25. Условие и ответ	34
3.30	Задание №26. Условие и ответ	35
3.31	Задание №27. Условие и ответ	36
3.32	Задание №28. Условие и ответ	37
3.33	Задание №29. Условие и ответ	37
3.34	Задание №30. Условие и ответ	38
3.35	Задание №31. Условие и ответ	39
3.36	Задание №31. Файл	39
3.37	Задание №32. Условие и ответ	40

3.38 Задание №33. Условие	40
3.39 Задание №33. Ответ	41
3.40 Задание №34. Условие и ответ	41
3.41 Задание №35. Условие и ответ	42
3.42 Задание №36. Условие и ответ	42

Список таблиц

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)

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

Верно решили 32 523 учащихся
Из всех попыток 69% верных

☒ Так точно!

☐ "q"

☐ "Q"

☒ ":", затем "q", затем "Enter"

☐ "Ctrl", затем "x"

☐ "q", затем "Enter"

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: 1 балл

Рис. 3.1: Задание №1. Условие и ответ

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

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 прочесть и про `e` 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

☒ Хорошая работа.

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](#).

- ☒ После 10 нажатий на W курсор окажется там же, где бы он был после 10 нажатий на w
- ☒ В этой строке 9 "слов" (word)
- ☐ В этой строке 5 "слов" (word)
- ☐ Нажимая только на w, нельзя переместить курсор на "."
- ☒ Нажимая только на W, нельзя переместить курсор на "."
- ☐ В этой строке 12 "слов" (word)

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **1 балл**

Рис. 3.2: Задание №2. Условие и ответ

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

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

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

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

Выберите все подходящие ответы из списка

☒ Верно. Так держать!

Верно решили 23 655 учащихся
Из всех попыток 16% 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.

- ☒ ddltree four four four five<Esc>
- ☐ d2dwywPp
- ☒ xxxxxxxxwywPp
- ☒ d2wwifour four <Esc>
- ☒ d2w\$bifour four <Esc>
- ☒ d2wwywPp

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: 1 балл

Рис. 3.3: Задание №3. Условие и ответ

4. 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

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** учащихся
Из всех попыток **57%** верных

:%s/Windows/Linux

Следующий шаг

Решить снова

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

Рис. 3.4: Задание №4. Условие и ответ

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

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

✓ Так точно!

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](#).

- ☐ Чтобы выйти из режима выделения, нужно ввести :q
- ☐ Режим выделения открывается при помощи команды :visual
- ☐ Режим выделения открывается из любого другого режима по нажатию "v"
- ☒ Режим выделения открывается из нормального режима по нажатию "v"
- ☒ Когда вы находитесь в режиме выделения, внизу редактора горит надпись – VISUAL – (или – ВИЗУАЛЬНЫЙ РЕЖИМ –)
- ☒ Выйти из режима выделения можно, нажав клавишу Esc два раза

Следующий шаг

Решить снова

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

Рис. 3.5: Задание №5. Условие и ответ

3.2 Скрипты на bash: основы

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

Надеемся, что вы разобрались, что одну оболочку (например, `sh`) можно запустить из другой оболочки (например, из `bash`).

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 попытаете при помощи стрелочек вверх/вниз перемещаться по истории набранных команд, то команды из какого набора(ов) будут появляться?

Выберите один вариант из списка

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

☒ Прекрасный ответ.

- ☐ Только из набора B
- ☐ Из наборов A и C
- ☐ Из наборов B и C
- ☒ Только из набора C
- ☐ Только из набора A

[Следующий шаг](#) [Решить снова](#)

[Ваши решения](#) Вы получили: **1 балл**

Рис. 3.6: Задание №6. Условие и ответ

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

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 окончании работы скрипта?

Выберите один вариант из списка

☒ Хорошие новости, верно!

Верно решили **29 905** учащихся
Из всех попыток **76%** верных

- ☐ `/home/bi/Desktop/file1.txt`
- ☒ `/home/bi/file1.txt`
- ☐ `/home/bi/Documents/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: Задание №7. Условие и ответ

3. 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.

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

☒ Всё правильно.

Верно решили **27 188** учащихся
Из всех попыток **25%** 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](#).

- ☐ 123variable
- ☒ VARiable
- ☐ var.i.able
- ☐ vari/able
- ☐ var@iable
- ☐ variab\$\$le
- ☒ __variable

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **1 балл**

Рис. 3.8: Задание №8. Условие и ответ

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

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`

✓ Прекрасный ответ.

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

Верно решили **25 053** учащихся
Из всех попыток **41%** верных

```
1 echo Arguments are: \$1=$1 \$2=$2
2
3
4
5
6
```

[Следующий шаг](#) [Решить снова](#)

[Ваши решения](#) Вы получили: **3 балла**

Рис. 3.9: 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.

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 вписать 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

☒ Правильно.

Верно решили **23 158** учащихся
Из всех попыток **16%** 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](#).

- ☒ `-z ""`
- ☒ `-n $0`
- ☐ `-z " "`
- ☐ `$var1 == $var2 && $var1 != $var2`
- ☒ `5 -ge 5`
- ☐ `-n $1`

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **1 балл**

Рис. 3.10: Задание №10. Условие и ответ

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

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

☒ Правильно, молодец!

Верно решили **25 138** учащихся
Из всех попыток **64%** верных

- ☒ Сначала four, потом four
- ☐ Сначала two, потом one
- ☐ Сначала one, потом two
- ☐ Сначала two, потом four

Следующий шаг

Решить снова

Ваши решения Вы получили: **1 балл**

Рис. 3.11: Задание №11. Условие и ответ

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

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](#).

Рис. 3.12: Задание №12. Условие

Напишите программу. Тестируется через stdin → stdout

✓ Правильно.

Верно решили **23 310** учащихся
Из всех попыток **38%** верных

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

```
1 arg=$1
2 case $arg in
3     0)
4         echo No students
5     ;;
6     1)
7         echo 1 student
8     ;;
9     2)
10        echo 2 students
11    ;;
12    3)
13        echo 3 students
14    ;;
15    4)
16        echo 4 students
17    ;;
18    *)
19        echo A lot of students
20 esac
21
22
23
24
25
```

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **3 балла**

Рис. 3.13: 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
```

4. Вопрос 13-й: Команда запустится 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.

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

☒ Здорово, всё верно.

Верно решили **24 582** учащихся
Из всех попыток **45%** верных

- ☐ 5 раз "start" и 2 раза "finish"
- ☐ 5 раз "start" и 5 раз "finish"
- ☒ 5 раз "start" и 4 раза "finish"
- ☐ 3 раза "start" и 2 раза "finish"

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **1 балл**

Рис. 3.14: Задание №13. Условие и ответ

5. 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.

Напишите скрипт на bash, который будет определять в какую возрастную группу попадают пользователи. При запуске скрипт 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:
Elena Petrovna
enter your age:
25
Elena Petrovna, your group is youth
enter your name:

bye
```

Рис. 3.15: Задание №14. Условие

Напишите программу. Тестируется через stdin → stdout

✓ Абсолютно точно.

Верно решили 21 670 учащихся
Из всех попыток 23% верных

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

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](#).

```
1 name=""
2 age=""
3 while true
4 do
5     echo enter your name:
6     read name
7     if [[ -z $name ]]
8     then
9         break
10    fi
11    echo enter your age:
12    read age
13    if [[ $age -eq 0 ]]
14    then
15        break
16    fi
17
18    if [[ $age -le 16 ]]
19    then
20        echo $name, your group is child
21    elif [[ $age -gt 16 ]] && [[ $age -le 25 ]]
22    then
23        echo $name, your group is youth
24    elif [[ $age -gt 25 ]]
25    then
26        echo $name, your group is adult
27    fi
28 done
29 echo bye
30
31
32
```

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: 4 балла

Рис. 3.16: 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 работает из-за неправильного операнда.

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

Какие(ая) из предложенных ниже инструкций увеличат 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

Верно решили **22 116** учащихся
Из всех попыток **20%** верных

✓ Здорово, всё 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.

- ☐ `let "a+=b"`
- ☐ `a=${a+$b}`
- ☒ `let "a=${a+$b}"`
- ☒ `let "a+=b"`
- ☒ `let a=a+b`

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **1 балл**

Рис. 3.17: Задание №15. Условие и ответ

2. Вопрос 16-й: Обратные одинарные кавычки позволяют выполнять команды внутри сценария оболочки, поэтому вывод будет равен пути `rwd`, указанном в файле.

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:

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

Что в этом случае выведет команда `echo` на экран?

Выберите один вариант из списка

Верно решили **23 677** учащихся
Из всех попыток **51%** верных

☒ Здорово, всё верно.

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

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **1 балл**

Рис. 3.18: Задание №16. Условие и ответ

3. 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.

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

Верно решили **21 426** учащихся
Из всех попыток **20%** верных

☒ Так точно!

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.

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

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **1 балл**

Рис. 3.19: Задание №17. Условие и ответ

4. 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.

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

✓ Правильно, молодец!

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 and 110

Следующий шаг

Решить снова

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

Рис. 3.20: Задание №18. Условие и ответ

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

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 вывести 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 отличие 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. 3.21: Задание №19. Условие

Напишите программу. Тестируется через stdin → stdout

✓ Правильно.

Верно решили 18 148 учащихся
Из всех попыток 35% верных

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

```
1 gcd()
2 {
3     local let "M+=${1}"
4     local let "N=${2}"
5     if [[ M -eq N ]]
6     then
7         return $M
8     elif [[ M -gt N ]]
9     then
10        let "r=$M-$N"
11        gcd $r $N
12    elif [[ M -lt N ]]
13    then
14        let "r=$N-$M"
15        gcd $M $r
16    fi
17 }
18
19 while true
20 do
21     read num1 num2
22     if [[ -z $num1 ]] || [[ -z $num2 ]]
23     then
24         echo bye
25         break
26     fi
27     gcd $num1 $num2
28     echo GCD is $?
29 done
30
31
32
```

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: 4 балла

Рис. 3.22: 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
    fi
}
```

```

    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

```

6. 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.

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 операнд (целое число), операция (одна из "+", "-", "*", "/", "%", "**") и второй операнд (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
```

т.к. вторая команда была **некорректной** (в ней всего один аргумент, т.к. нет пробелов между числами и операцией, а единственная допустимая команда из одного аргумента это "exit").

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

Рис. 3.23: Задание №20. Условие

Напишите программу. Тестируется через stdin → stdout

✓ Верно. Так держать!

Верно решили 16 980 учащихся
Из всех попыток 36% верных

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

```
1 while true
2 do
3     read num1 op num2
4     if [[ $num1 == "exit" ]]
5     then
6         echo bye
7         break
8     elif [[ "$num1" =~ "^[-0-9]+$" && "$num2" =~ "^[-0-9]+$" ]]
9     then
10        echo error
11        break
12    fi
13    case $op in
14        "+")
15        let "res=$num1+$num2"
16        ;;
17        "-")
18        let "res=$num1-$num2"
19        ;;
20        "=")
21        let "res=$num1*$num2"
22        ;;
23        "/"")
24        let "res=$num1/$num2"
25        ;;
26        "%")
27        let "res=$num1%$num2"
28        ;;
29        "**")
30        let "res=$num1**$num2"
31        ;;
32        *)
33        echo error
34        break
35    esac
36    echo $res
37 done
38
39
40
```

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: 5 баллов

Рис. 3.24: 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.

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

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

✓ Всё правильно.

- ☒ STARS.txt
- ☐ Eddard_Stark_biography.txt
- ☒ Star_Wars.avi
- ☐ stardust.mpeg
- ☐ star_trek OST.mp3

Следующий шаг Решить снова

Ваши решения Вы получили: **1 балл**

👍 785 🗨 244 Шаг 3 Следующий шаг >

Рис. 3.25: Задание №21. Условие и ответ

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

Задание на понимание работы опций `-path` и `-name` команды `find`. Отметьте **все верные** утверждения из перечисленных ниже.

Выберите все подходящие ответы из списка

Верно решили **18 450** учащихся
Из всех попыток **22%** верных

✓ 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](#).

- ☐ Опции `-path` и `-name` всегда работают одинаково
- ☐ Опция `-path` аналогична `-name`, но игнорирует размер букв (строчные/прописные) в имени файла
- ☐ Если заменить в команде поиска `-name`, на `-path`, то результат поиска всегда останется неизменным
- ☐ Опция `-path` используется только для поиска директорий, а `-name` только для поиска файлов
- ☒ Если заменить в команде поиска `-name`, на `-path`, то результат поиска иногда может остаться таким же

Следующий шаг Решить снова

Ваши решения Вы получили: **1 балл**

Рис. 3.26: Задание №22. Условие и ответ

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

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

Верно решили 20 711 учащихся
Из всех попыток 41% верных

☒ Здорово, всё верно.

☐ Только file1
☐ Все кроме file1
☐ Все кроме file2
☒ Все кроме file3
☐ Только file2

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: 1 балл

Рис. 3.27: Задание №23. Условие и ответ

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

- -A n – Вывод n строк контекста после группы найденных вхождений;
- -B n – Вывод n строк контекста перед 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.

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

Верно решили **20 237** учащихся
Из всех попыток **41%** верных

✓ Отличное решение!

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

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **1 балл**

Рис. 3.28: Задание №24. Условие и ответ

5. 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`.

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

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

✓ Так точно!

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
- ☒ Mac OS X, Windows, Ubuntu
- ☒ Hmm, XKLubuntu
- ☒ Lubuntu is better than Ubuntu
- ☐ Lubuntu is better than Windows
- ☐ Uuuubuntu!

Следующий шаг

Решить снова

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

Рис. 3.29: Задание №25. Условие и ответ

6. Вопрос 26-й: Без опции `-n` происходит автоматический вывод всех строк, при этом `r` в конце выражения указывает на то, что дополнительно должны

быть выведены строки, удовлетворяющие шаблону. Произойдет удвоение строк.

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

Выберите один вариант из списка

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

☒ Правильно.

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

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: **1 балл**

Рис. 3.30: Задание №26. Условие и ответ

7. 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;

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 с "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 повторов (т.е. *, +, ? и []). Однако эту "несовместимость" легко исправить указав при помощи "(" и ")" какой из символов к чему относится! Например, регулярное выражения `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

✓ Хорошие новости, верно!

Верно решили 16 632 учащихся
Из всех попыток 34% 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.

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

Следующий шаг

Решить снова

Ваши решения Вы получили: 3 балла

Рис. 3.31: 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.

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

☒ Отличное решение!

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

- ☐ -s, --show-plots-after-exit
- ☒ -p, --persist
- ☐ Такой опции не существует
- ☐ -raise

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **1 балл**

Рис. 3.32: Задание №28. Условие и ответ

2. 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.

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

☒ Абсолютно точно.

Верно решили **17 975** учащихся
Из всех попыток **32%** 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](#).

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

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **1 балл**

Рис. 3.33: Задание №29. Условие и ответ

3. Вопрос 30-й: Установим деления командой set, укажем ось Oх с помощью xtics и введем через запятую точки в формате ("название", значение).

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 пишете gnuplot-скрипт и у вас в нем есть три переменные `x1`, `x2`, `x3`, в которых записаны координаты важных точек по оси OX (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

✓ Отличное решение!

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)
```

Следующий шаг

Решить снова

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

Рис. 3.34: Задание №30. Условие и ответ

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

- отразится зеркально, если поставить отрицание перед функцией и раскрыть скобки: `plot -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`

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

✓

Здорово, всё верно.

Верно решили **12 854** учащихся
Из всех попыток **47%** верных

move.rot (93 bytes)

Следующий шаг

Решить снова

[Ваши решения](#) Вы получили: **3 балла**

Рис. 3.35: Задание №31. Условие и ответ

```
*move (1).rot – Блокнот
Файл  Правка  Формат  Вид  Справка
a=a+1
zrot=(zrot+350)%360
set view xrot,zrot
splot -x**2-y**2
pause 0.1
if (a<50) reread
|
```

Рис. 3.36: Задание №31. Файл

3.7 Разное

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

Какая команда(ы) установят файлу `file.txt` права доступа `rw-rw-r--`, если изначально у него были права `r--r--r--`.
Укажите **все верные** варианты ответа!

Примечание: запись вида `команда1; команда2; команда3` означает, что в терминале последовательно выполнялись все три команды (сначала `команда1`, затем `команда2` и, наконец, `команда3`).

Выберите все подходящие ответы из списка

Верно решили **16 484** учащихся
Из всех попыток **21%** верных

✓ 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](#).

☒ `chmod u+wx file.txt; chmod g+w file.txt`
☐ `chmod 467 file.txt`
☒ `chmod 764 file.txt`
☐ `chmod u-wx file.txt; chmod g-w file.txt`
☒ `chmod a+wx file.txt; chmod o-wx file.txt; chmod g-x file.txt`
☐ `chmod rw-rw-r-- file.txt`

Следующий шаг Решить снова

Ваши решения Вы получили: **1 балл**

Рис. 3.37: Задание №32. Условие и ответ

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

Предположим вы использовали команду `sudo` для создания директории `dir`. По умолчанию для `dir` были выставлены права доступа `rw-r-xr-x` (владелец `root`, группа `root`). Таким образом никто кроме пользователя `root` не может ничего записывать в эту директорию, например, не может создавать файлы в ней.

После выполнения какой команды `user` из группы `group` всё-таки сможет создать файл внутри `dir`? Укажите **все верные** варианты ответов!

Примечание: считаем, что все команды выполняются от имени `user`, если явно не указано, что команда выполнена с `sudo`.

Примечание 2: мы выбрали пример с директорией, а не с файлом не случайно. Дело в том, что если создать при помощи `sudo` файл с правами `rw-r--r--` в директории, которая принадлежит пользователю, то возникнет любопытная ситуация. С одной стороны пользователь может удалить этот файл (т.к. ему разрешено удалять **все** файлы внутри его директории) и может прочитать его содержимое (т.к. право `"r"` у файла установлено для всех), с другой стороны он не может этот файл редактировать (т.к. право `"w"` у файла есть только для `root`). При этом некоторые "умные" редакторы, например, `vim` позволят даже редактировать этот файл, но сделают они это своеобразно: через удаление оригинала и создание копии уже с нужными правами (удалять мы можем, а раз можем читать, то и копию создать не сложно). Итого получается, что несмотря на права `rw-r--r--`, пользователь может сделать с этим файлом почти всё что угодно!

В случае же, когда речь идет о директории созданной `root`, ситуация будет проще: пользователь сможет смотреть её содержимое (у него есть право `"r"`), но удалять и создавать файлы в ней не сможет (права `"w"` у него нет). Важно отметить, что директории в `Linux` это в каком-то смысле *файлы*. Содержимое такого "файла" – это записи о файлах и поддиректориях этой директории (грубо говоря их названия). Таким образом, право `"r"` у директории дает возможность просматривать "записи", т.е. просматривать её состав. Право `"w"` у директории дает возможность удалять/добавлять новые "записи", т.е. удалять/создавать файлы/поддиректории в ней.

На самом деле и это еще не всё. Существует так называемый *sticky bit* (атрибут файла или директории), выставление которого меняет описанное выше поведение. Файлы (или директории) с таким атрибутом сможет удалить только их владелец вне зависимости от прав, установленных у директории, в которой эти файлы (или директории) лежат!

Отдельное спасибо слушателю курса **Alexey Antipovsky** за помощь в оформлении **Примечания 2!**

Рис. 3.38: Задание №33. Условие

Выберите все подходящие ответы из списка

Верно решили 14 683 учащихся
Из всех попыток 15% верных

✓ 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](#).

☐ sudo chmod o+x dir
☒ sudo chmod a+w dir
☒ sudo chmod o+w dir
☐ sudo chown :group dir
☐ chown user.group dir
☐ chmod o+w dir

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: 1 балл

Рис. 3.39: Задание №33. Ответ

3. Вопрос 34-й: Откроем справку по команде `wc` и посмотрим 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;

Отметьте какие характеристики файла можно посчитать с использованием команды `wc`.

Выберите все подходящие ответы из списка

Верно решили 17 158 учащихся
Из всех попыток 21% верных

✓ Так точно!

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](#).

☒ Количество символов
☒ Длину самой длинной строки
☐ Количество предложений
☐ Количество определенных букв (например, количество букв "А")
☒ Количество слов

Следующий шаг Решить снова

[Ваши решения](#) Вы получили: 1 балл

Рис. 3.40: Задание №34. Условие и ответ

4. Вопрос 35-й: Выведем команду `du`, анализирующую занимаемое файло-

вое пространство, с двумя ключами `-h` или `--human-readable` и `-s` или `--summarize`.

Впишите в форму ниже команду, которая выведет сколько места на диске занимает текущая директория (при этом **размер** нужно вывести **в удобном для чтения формате** (например, вместо `2848 байт` надо выводить `2.8K`) и **больше** на экран выводить **ничего не нужно**). В команде указывайте **только необходимые** для выполнения задания **опции и аргументы**, лишних опций указывать не нужно!

Пример: если в текущей директории есть два файла по `800 Кбайт` и две поддиректории в каждой из которых лежит по файлу в `400 Кбайт`, то загаданная команда должна вывести на экран одно число: `2.4M` (также на экране может быть выведен еще и символ `"`, обозначающий, что это размер именно текущей директории).

Напишите текст

✓ Отлично!

Верно решил **16 381** учащихся
Из всех попыток **53%** верных

`du -h -s`

Следующий шаг

Решить снова

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

Рис. 3.41: Задание №35. Условие и ответ

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

Впишите в форму ниже максимально короткую команду (т.е. в которой минимально возможное число символов), которая позволит создать в текущей директории 3 поддиректории с именами `dir1`, `dir2`, `dir3`.

Если вы придумали команду, которая выполняет эту задачу, а система проверки сообщает вам `"Incorrect"/"Неверно"`, то скорее всего вы придумали не самую короткую команду из возможных!

Напишите текст

✓ Верно. Так держать!

Верно решили **16 720** учащихся
Из всех попыток **40%** верных

`mkdir dir{1..3}`

Следующий шаг

Решить снова

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

Рис. 3.42: 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>.