

Отчёт по прохождению курса третий этап

Абронина Алиса Кирилловна

Содержание

1 Продвинутые темы	4
1.1 3.1 Текстовый редактор vim	4
1.2 3.2 Скрипты на bash: основы	6
1.3 3.3 Скрипты на bash: ветвлени и циклы	9
1.4 3.4 Скрипты на bash: разное	12
1.5 3.5 Продвинутый поиск и редактирование	16
1.6 3.6 Строим графики в gnuplot	19
1.7 3.7 Разное	21

List of Figures

1.1 Задание 1	4
1.2 Задание 2	5
1.3 Задание 3	5
1.4 Задание 4	6
1.5 Задание 5	6
1.6 Задание 1	7
1.7 Задание 2	7
1.8 Задание 3	8
1.9 Задание 4	8
1.10 Задание 1	9
1.11 Задание 2	10
1.12 Задание 3	10
1.13 Задание 4	11
1.14 Задание 5	12
1.15 Задание 1	13
1.16 Задание 2	13
1.17 Задание 3	14
1.18 Задание 4	14
1.19 Задание 5	15
1.20 Задание 6	15
1.21 Задание 1	16
1.22 Задание 2	16
1.23 Задание 3	17
1.24 Задание 4	17
1.25 Задание 5	18
1.26 Задание 6	18
1.27 Задание 7	19
1.28 Задание 1	19
1.29 Задание 2	20
1.30 Задание 3	20
1.31 Задание 4	21
1.32 Задание 1	21
1.33 Задание 2	22
1.34 Задание 3	22
1.35 Задание 4	23
1.36 Задание 5	23

1 Продвинутые темы

1.1 3.1 Текстовый редактор vim

: переводит в командный режим; q команда вывода; Enter выполнение команды

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

Хорошая работа.

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

[Следующий шаг](#) [Решить снова](#)

Верно решили 32 523 учащихся
Из всех попыток 69% верных

Ваши решения Вы получили: 1 балл

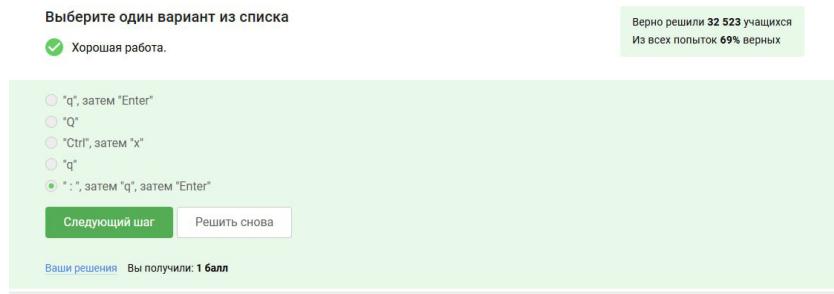


Figure 1.1: Задание 1

word - разбивает по буквам, цифрам, знакам; WORD - разбивает только по пробелам -> Strange_ TEXT is here. 2=2 YES! -> 5 WORD

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 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

Верно. Так держать!

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 25 385 учащихся
Из всех попыток 20% верных

- В этой строке 12 "слов" (word)
- Нажимая только на w, нельзя переместить курсор на " "
- Чтобы попасть в конец строки, нужно совершить больше нажатий на W, чем на w
- Чтобы попасть в конец строки, нужно одинаковое число нажатий, что на W, что на w
- В этой строке 5 "больших слов" (WORD)
- В этой строке 5 "слов" (word)

[Следующий шаг](#)

[Решить снова](#)

Figure 1.2: Задание 2

d2w - удалить два слова; ww - перейти на слово three; yw - yank (копировать)
four; P,p - вставка до и после курсора; i - вставка текста вручную

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

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 23 655 учащихся
Из всех попыток 16% верных

- d2w\$bfour four <Esc>
- d2wwwyPp
- x2wwwywPp
- d2wwwywPp
- d2wwwywpp
- d2wwwifour four <Esc>

[Следующий шаг](#)

[Решить снова](#)

Ваши решения Вы получили: 1 балл

Figure 1.3: Задание 3

делала все по заданию

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

Правильно.

Верно решил 24 631 учащийся
Из всех попыток 57% верных

```
:%s/Windows/Linux
```

Следующий шаг Решить снова

Ваше письмо Вы получили: ...

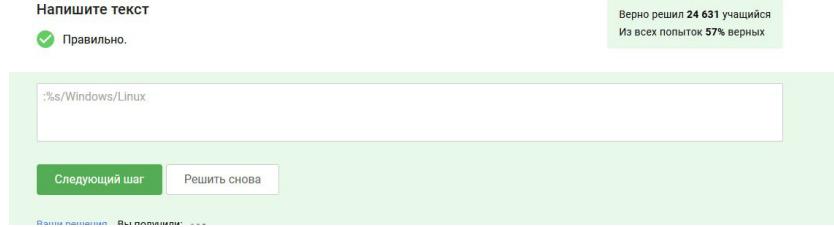


Figure 1.4: Задание 4

v - войти в Visual mode; d- удалить выделенное; u - копировать

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

Абсолютно 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 свое решение с другими на [форуме решений](#).

В режиме выделения можно использовать команды перемещения (например, W, e, \$, и др.)

Режим выделения открывается из любого другого режима по нажатию "v"

Чтобы выйти из режима выделения, нужно ввести :q

Режим выделения открывается при помощи команды :visual

Режим выделения открывается из нормального режима по нажатию "v"

В режиме выделения можно использовать команды d (удалить) и u (скопировать)

Следующий шаг Решить снова

Ваши решения Вы получили: ...

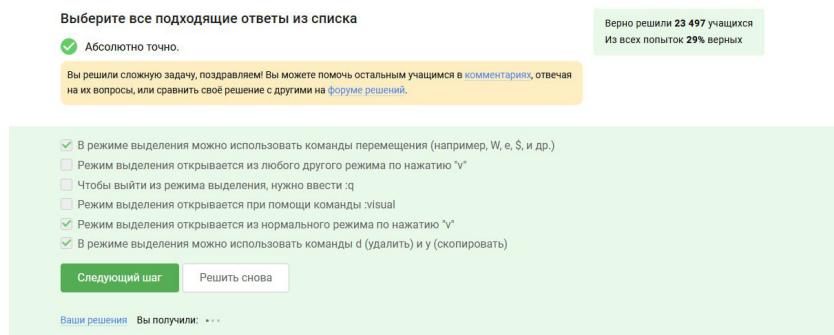


Figure 1.5: Задание 5

1.2 3.2 Скрипты на bash: основы

каждая оболочка имеет свою локальную историю; последний активный bash сохраняет только свои 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 попробуете при помощи стрелочек вверх/вниз перемещаться по истории набранных команд, то команды из какого набора(ов) будут появляться?

Выберите один вариант из списка

Отличное решение!

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

- Из наборов А и С
- Только из набора С
- Никакие команды появляться не будут
- Из наборов В и С
- Только из набора А

Следующий шаг **Решить снова**

Ваши решения Вы получили: **1 балл**

Figure 1.6: Задание 1

сначала переход `/home/bi`, там создается файл, `cd` на Desktop уже после создания

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

Правильно, молодец!

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

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

Следующий шаг **Решить снова**

Ваши решения Вы получили: **1 балл**

Figure 1.7: Задание 2

переменные начинаются с буквы или `_`, далее цифры и буквы; остальные варианты содержат недопустимые символы

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

Правильно, молодец!

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

vari/able
 variab\$\$le
 vari-able
 123variable
 vari.i.able
 var i able
 variable123

Следующий шаг **Решить снова**

Ваши решения Ваши решения: Вы получили: 1 балл

Верно решили 27 188 учащихся
 Из всех попыток 25% верных

Figure 1.8: Задание 3

Скрипт выводит аргументы в нужном формате, экранируя \$1 и \$2, чтобы они отображались как текст

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

Правильно.

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

```
1 var1=$1
2 var2=$2
3
4 echo "Arguments are: \$1=$var1 \$2=$var2"
5
6
7
8
9
```

Следующий шаг **Решить снова**

Верно решили 25 053 учащихся
 Из всех попыток 41% верных

Figure 1.9: Задание 4

1.3 3.3 Скрипты на bash: ветвления и циклы

`$#` - кол-во аргументов(всегда ≥ 0), `5 -ge 5` - истина `-s $0` - файл скрипта существует и не пуст

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, молодец!

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](#).

- `$# -ge 0`
- `5 -ge 5`
- `$var1 == $var2 || $var1 != $var2`
- `-s $0`
- `$var1 == $var2 && $var1 != $var2`
- `$# -gt 0`

Figure 1.10: Задание 1

первое условие – ложь, второе – ложь, третье – ложь, выполняется `echo four` потому `var =5` все условие ложные и снова `four`

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

Прекрасный ответ.

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

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

Следующий шаг

Решить снова

Figure 1.11: Задание 2

если аргумент равен 1, выводится 1 student, если от 2 до 4 N students, если 5 или больше, выводится A lot of students. Скрипт реализует это через case и переменную res, как требует задание

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

Правильно.

Верно решили **23 310** учащихся

Из всех попыток **38%** верных

```
1 #!/bin/bash
2 v@student #задаем общую часть в каждом выводе - слово "student"
3 case $1 in
4 *) res="$1 ${v}s"; #res - это результат для вывода
5 *) res="$1 ${v}";;
6 *[2-4]) res="$1 ${v}s";;
7 *) res="A lot of ${v}s";;
8 esac
9 echo "$res" #вывести результат
10 ..
```

Figure 1.12: Задание 3

перебираются строки a,b c_d всего 5 итераций: каждая итерация выводит start,

если str > "c" выполняется continue, пропуская echo "finish", только c_d > c поэтому continue срабатывает 1 раз

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

Всё правильно.

Верно решили 24 582 учащихся
Из всех попыток 45% верных

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

[Следующий шаг](#)

[Решить снова](#)

[Ваши решения](#) Вы получили: 1 балл

Figure 1.13: Задание 4

проверка на пустой ввод и неккоректный возраст есть, группы определяются правильно по условиям, программа работает до тех пор, пока пользователь не выводит пустое имя или неккоректный возраст, вывод соответствует требованиям

Напишите программу. Тестируется через stdin → stdout

✓ Хорошие новости, верно!

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 свое решение с другими на [форуме решений](#).

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

Figure 1.14: Задание 5

1.4 3.4 Скрипты на bash: разное

let “a+=b” правильно, но не отмечено, возможно баг; a=a+b записывать строку, ане сумму; let a = a + b синтаксическая ошибка; a+=\$b пробел делает команду некорректной

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

Какие(а) из предложенных ниже инструкций увеличат значение переменной `a` на значение переменной `b`? Например, если `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

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

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

[Следующий шаг](#)

[Решить снова](#)

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

Figure 1.15: Задание 1

после `cd /home/bi` `pwd` покажет эту директорию

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.

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

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

[Следующий шаг](#)

[Решить снова](#)

[Ваши решения](#) Вы получили: **++**

Figure 1.16: Задание 2

program запускает команду `$?` содержит код невозврата, перенаправляет вывод

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

Отлично!

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 21 426 учащихся

Из всех попыток 20% верных

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

[Следующий шаг](#)

[Решить снова](#)

Ваши решения Вы получили: 1 балл

Figure 1.17: Задание 3

делала все по заданию

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 пределает ровно то, что указано в задании и поизвольно сверить свой ответ с тем, что он выдаст на экран.

Напишите текст

Абсолютно 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](#).

Верно решили 20 009 учащихся

Из всех попыток 28% верных

counters are and 110

[Следующий шаг](#)

[Решить снова](#)

Figure 1.18: Задание 4

алгоритм Евклида реализован корректно(с вычитание, ане остатком), условия выхода из функции - проверка равенста, бесконечный цикл с возможность выхода реализован правильно, проверка на завершение работы реализована через пустой ввод

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.

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

✓ Верно. Так держать!

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

Верно решили 18 148 учащихся
Из всех попыток 35% верных

```

1 #!/bin/bash
2
3 # Функция нахождения НОД по алгоритму Евклида
4 gcd() {
5     local m=$1
6     local n=$2
7     if [ "$m" -eq "$n" ]; then
8         echo "GCD is $m"
9     elif [ "$m" -gt "$n" ]; then
10        gcd ${((m - n))} $n
11    else
12        gcd $m ${((n - m))}
13    fi
14 }
15
16 # Основной бесконечный цикл
17 while true; do
18     read m n
19     if [ -z "$m" ]; then
20         echo "bye"
21         break
22     fi
23     gcd "$m" "$n"
24 done
25

```

Figure 1.19: Задание 5

есть коректный выход по команде `exit`, есть проверка на валидность чисел через регулярные выражения, есть базовые арифметические операции реальзованные через `case` скрипт завершает работу при любой ошибке

Подсказка: в случае проблем с решением задачи, обратите внимание на наши рекомендации по написанию скриптов.

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

✓ Так точно!

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

Верно решили 16 980 учащихся
Из всех попыток 36% верных

```

1
2 #!/bin/bash
3 while [[ $True ]]
4 do
5     read birinchi amal ikkinchi
6     if [[ $birinchi == "exit" ]]
7     then
8         echo "bye"
9         break
10    elif [[ "$birinchi" =~ ^[0-9]+\$" && "$ikkinchi" =~ ^[0-9]+\$" ]]
11    then
12        echo "error"
13        break
14    else
15        case $amal in
16            "+") let "result = birinchi + ikkinchi";;
17            "-") let "result = birinchi - ikkinchi";;
18            "/") let "result = birinchi / ikkinchi";;
19            "*") let "result = birinchi * ikkinchi";;
20            "%") let "result = birinchi % ikkinchi";;
21            "**") let "result = birinchi ** ikkinchi";;
22        *) echo "error" ; break ;;
23        esac
24        echo "$result"
25    fi
26 done
27

```

Figure 1.20: Задание 6

1.5 3.5 Продвинутый поиск и редактирование

name чувствителен к регистру, iname - игнорирует регистр

Пусть в директории /home/bi лежат файлы Star_Wars.avi, star_trek_OST.mp3, STARS.txt, stardust.mpeg, Eddard_Stark_biology.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
✓ Отличное решение!

star_trek_OST.mp3
 Star_Wars.avi
 Eddard_Stark_biology.txt
 STARS.txt
 stardust.mpeg

Верно решили 20 547 учащихся
Из всех попыток 36% верных

Следующий шаг Решить снова

Ваши решения Вы получили: 1 балл

Figure 1.21: Задание 1

path проверяет весь путь , name только имя файла проверяет

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

Выберите все подходящие ответы из списка
✓ Верно. Так держать!

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](#).

В некоторых случаях `find c -name` найдет больше файлов, чем `find c` с таким же запросом, но `c-path`
 Опция `-path` используется только для поиска директорий, а `-name` только для поиска файлов
 Опции `-path` и `-name` всегда работают одинаково
 Опция `-path` аналогична `-name`, но игнорирует размер букв (строчные/прописные) в имени файла
 В некоторых случаях `find c -name` найдет меньше файлов, чем `find c` с таким же запросом, но `c-path`

Верно решили 18 450 учащихся
Из всех попыток 22% верных

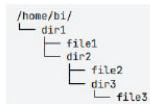
Следующий шаг Решить снова

Ваши решения Вы получили: 1 балл

Figure 1.22: Задание 2

mindpath 2 - ищет только начиная со второго уровня(то есть первый исключается), а есть третий у него глубина 4

Предположим, что в директории `/home/b1/` есть следующая структура файлов и поддиректорий:



Какие(ой) из трех файлов (`file1`, `file2`, `file3`) будут найдены по команде `find /home/b1 -mindepth 2 -maxdepth 3 -name "file*"`?

Выберите один вариант из списка

Верно. Так держать!

Верно решили 20 711 учащихся
Из всех попыток 41% верных

- Только `file3`
- Все кроме `file2`
- Все кроме `file3`
- Только `file2`
- Все кроме `file1`

[Следующий шаг](#)

[Решить снова](#)

Ваши решения Вы получили: 1 балл

Figure 1.23: Задание 3

-A 1, -B 1, -C 1 добавляют дополнительные строки вокруг совпадений. Тк каждая строка совпадение, результат будет одинаково полным в любом случае

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

Правильно, молодец!

Верно решили 20 237 учащихся
Из всех попыток 41% верных

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

[Следующий шаг](#)

[Решить снова](#)

Ваши решения Вы получили: 1 балл

Figure 1.24: Задание 4

в начале команды - 1 символ или ничего дальше ubuntu с любым регистром \$
конец строки

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

Выберите все подходящие ответы из списка

Абсолютно 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](#).

Uuuubuntul
 Kubuntu
 Lubuntu is better than Windows
 Well, xubuntu is OK
 The best OS is Xubuntu
 Mac OS X 10.9, Windows XP, Ubuntu 12.04

[Следующий шаг](#) [Решить снова](#)

Ваши решения Вы получили: 2 балла

Верно решили 18 768 учащихся
Из всех попыток 23% верных

Figure 1.25: Задание 5

`sed` по умолчанию выводит всю строку, `p` - означает повторно печатать строки

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

Выберите один вариант из списка

Верно.

Каждая строчка будет выведена два раза
 На экран ничего не напечатается
 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

Верно решили 19 784 учащихся
Из всех попыток 39% верных

Figure 1.26: Задание 6

делала все по заданию

- 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 `! -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 эта "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

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](#).

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

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

[Следующий шаг](#)

[Решить снова](#)

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

Figure 1.27: Задание 7

1.6 3.6 Строим графики в gnuplot

опция `r` сохраняет окно графика открытым после завершения `gnuplot`

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

Абсолютно точно.

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

- Графики и так не закрываются автоматически при закрытии `gnuplot`!
- Такой опции не существует
- `-raise`
- `-p, --persist`

[Следующий шаг](#)

[Решить снова](#)

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

Figure 1.28: Задание 1

команда `set key autotitle columnhead` говорит `gnuplot` использовать газоловки столбцов как названия но в `data/csv` нет заголовков

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

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](#).

Верно решили 17 975 учащихся
Из всех попыток 32% верных

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

[Следующий шаг](#)

[Решить снова](#)

Ваши решения Вы получили: 1 балл

Figure 1.29: Задание 2

делала все по заданию

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.

Припишите 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

Абсолютно точно.

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](#)

Figure 1.30: Задание 3

делала все по заданию

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

Отлично!

Верно решили 12 854 учащихся
Из всех попыток 47% верных

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

Следующий шаг

Решить снова

Figure 1.31: Задание 4

1.7 3.7 Разное

`rwxrw-r- =764 (в восьмеричной СС)`

Какая команда(ы) установят файлу `file.txt` права доступа `rwxrw-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 выполнились все три команды (сначала `команда1`, затем `команда2` и, наконец, `команда3`).

Выберите все подходящие ответы из списка

Отлично!

Верно решили 16 484 учащихся
Из всех попыток 21% 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](#).

chmod u+w file.txt; chmod g+w file.txt

chmod u-wx file.txt; chmod g-w file.txt

chmod 777 file.txt

chmod 467 file.txt

chmod 764 file.txt

chmod o-wx file.txt; chmod g-x file.txt; chmod a+wx file.txt

Следующий шаг

Решить снова

Ваши решения Вы получили: 1 балл

Figure 1.32: Задание 1

`chmod +w dir` - дает право на изменение содержимого, `chown` - передает права владельца, что дает полные возможности в каталоге

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 `rwt-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 установлено 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 `rwt-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 назначении). 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 описание выше поведение: Файлы (или директории) с таким атрибутом смогут удалять 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!

Выберите все подходящие ответы из списка

Верно. Так держать!

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 o+r dir
 sudo chown :group dir
 chown user:group dir
 sudo chmod g+w dir
 sudo chown user dir

[Следующий шаг](#) [Решить снова](#)

Верно решили 14 683 учащихся
Из всех попыток 15% верных

Figure 1.33: Задание 2

wc считает: строки, слова, байсты, длину самой длинной строки

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

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](#).

Количество предложений
 Количество слов
 Длину самой длинной строки
 Количество строк
 Размер файла в байтах

[Следующий шаг](#) [Решить снова](#)

Ваши решения Вы получили: 1 балл

Верно |
Из всех

Figure 1.34: Задание 3

делала по заданию

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 2.8К) 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: 2.4М (также на экране может быть выведен еще и символ "", обозначающий, что это размер именно текущей директории).

Напишите текст

✓ Прекрасный ответ.

Верно решил 16 381 учащийся
Из всех попыток 53% верных

```
du -h -s
```

Следующий шаг

Решить снова

Ваши решения Вы получили: 2 балла

Figure 1.35: Задание 4

делалал по заданию

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

✓ Отличное решение!

Верно решили 16 720 учащихся
Из всех попыток 40% верных

```
mkdir dir{1..3}
```

Следующий шаг

Решить снова

Ваши решения Вы получили: 2 балла

Figure 1.36: Задание 5