

Слово — это последовательность из букв (русских или английских), внутри которой могут быть дефисы.

На вход даётся текст, посчитайте, сколько в нём слов.

PS. Задача решается в одну строчку.

Ввод	Вывод
Он --- серо-буро-малиновая редиска!! >>>:-> А не кот. www.kot.ru	9

```
anex13@anextmshsrv:~/Мисько/hw13$ cat test.txt | grep -Eo '[a-zA-Za-яА-Я-]+\b' --count
3
anex13@anextmshsrv:~/Мисько/hw13$ cat test.txt | grep -Eo '[a-zA-Za-яА-Я-]+\b'
Он
серо-буро-малтновая
редиска
А
не
кот
www
kot
ru
anex13@anextmshsrv:~/Мисько/hw13$ cat test.txt | grep -Eo '[a-zA-Za-яА-Я-]+\b' | wc -l
9
anex13@anextmshsrv:~/Мисько/hw13$
```

Ps не оч понятное поведение каунта хотя вероятно он считает строки в тексте в которых есть матчи.

1. Добавить ко всем файлам формата log timestamp в качестве названия.

Должно получиться filename_{timestamp}.log

Для всех файлов с расширением Ру добавьте в конец названия хэш коммита.

Создадим тестовое окружение

```
.
├── code1.py
├── code4.py
├── log1.log
├── log2.log
├── subdir
│   ├── code2.py
│   ├── code3.py
│   └── log3.log
```

```

GNU nano 6.2 script.sh
#!/bin/bash

dir=${1:-"./"}
for file in $(find "$dir" -type f)
do
    if [[ "$file" == *.py ]]
    then
        com=`git log $file | grep -m 1 commit | awk -F ' ' '{printf $2}'`
        echo "${file::-3}_${com::7}.py"
    elif [[ "$file" == *.log ]]
    then
        ts=`stat -c %w $file | awk -F ' ' '{printf $1}'`
        echo "${file::-4}_${ts}.log"
    fi
done

```

проверяем

```

anex13@anextmshwsrv:~/Мисько/hw13/testdir$ ./script.sh
./log1_2024-06-03.log
./subdir/log3_2024-06-03.log
./subdir/code3_fa40344.py
./subdir/code2_fa40344.py
./log2_2024-06-03.log
./code1_fa40344.py
./code4_5c97ef0.py

```

Теперь меняем echo.... на mv \$file

```

#!/bin/bash

dir=${1:-"./"}
for file in $(find "$dir" -type f)
do
    if [[ "$file" == *.py ]]
    then
        com=`git log $file | grep -m 1 commit | awk -F ' ' '{printf $2}'`
        mv $file ${file::-3}_${com::7}.py
    elif [[ "$file" == *.log ]]
    then
        ts=`stat -c %w $file | awk -F ' ' '{printf $1}'`
        mv $file ${file::-4}_${ts}.log
    fi
done

```

```

anex13@anextmshwsrv:~/Мисько/hw13/testdir$ tree
.
├── code1_fa40344.py
├── code4_5c97ef0.py
├── log1_2024-06-03.log
├── log2_2024-06-03.log
├── script.sh
└── subdir
    ├── code2_fa40344.py
    ├── code3_fa40344.py
    └── log3_2024-06-03.log

```

2. Довольно распространённая ошибка — это повтор слова.
Вот в предыдущем предложении такая допущена. Необходимо
исправить каждый такой повтор (слово, один или несколько
пробельных символов, и снова то же слово)

Опционально:

Допустимый формат e-mail адреса регулируется стандартом RFC 5322.

Если говорить вкратце, то e-mail состоит из одного
символа @ (at-символ или собака), текста до собаки (Local-part) и текста
после собаки (Domain part). Вообще в адресе может быть всякий беспредел
(вкратце можно прочесть о нём в википедии). Довольно странные штуки
могут быть валидным адресом, например:

```
"very.(),,;<>[]\".VERY.\"very@\\ \"very\".unusual"@[IPv6:2001:db8::1]
```

```
"()<>[:;,@\\\"!#$%&'-/=?^_`{|}~.a"@ (comment)example
```

Но большинство почтовых сервисов такой ад и вакханалию не допускают. И
мы тоже не будем :)

Будем рассматривать только адреса, имя которых состоит из не более, чем 64
латинских букв, цифр и символов '._+-', а домен — из не более, чем 255
латинских букв, цифр и символов '._-'. Ни Local-part, ни Domain part не может
начинаться или заканчиваться на '+-', а ещё в адресе не может быть более
одной точки подряд.

Кстати, полезно знать, что часть имени после символа + игнорируется,
поэтому можно использовать синонимы своего адреса
(например, shashkov+spam@179.ru и shashkov+vk@179.ru), для того, чтобы
упростить себе сортировку почты. (Правда не все сайты позволяют
использовать "+", увы)

На вход даётся текст. Необходимо вывести все e-mail адреса, которые в нём
встречаются. В общем виде задача достаточно сложная, поэтому у нас будет 3
ограничения:

две точки внутри адреса не встречаются;

две собаки внутри адреса не встречаются;

считаем, что e-mail может быть частью «слова», то есть в boo@ya_ru мы
видим адрес boo@ya, а в fooNoboo@ya.ru видим boo@ya.ru.

PS. Совсем не обязательно делать все проверки только регулярками.

Регулярные выражения — это просто инструмент, который делает часть задач
простыми. Не нужно делать их назад сложными :)

Ввод	Вывод
<p>Иван Иванович!</p> <p>Нужен ответ на письмо от ivanoff@ivan-chai.ru.</p> <p>Не забудьте поставить в копию serge'o-lupin@mail.ru- это важно.</p>	<p>ivanoff@ivan-chai.ru</p> <p>serge'o-lupin@mail.ru</p>
<p>NO: foo.@ya.ru, foo@.ya.ru</p> <p>PARTLY: boo@ya_ru, -boo@ya.ru-, foo%boo@ya.ru</p>	<p>boo@ya</p> <p>boo@ya.ru</p> <p>boo@ya.ru</p>