

1. Вам в руки попал набор заклинаний, не терпится их испробовать. Но, прежде чем приступить к экспериментам, нужно вычеркнуть особо опасные. Вы знаете, что к особо опасным относятся заклинания, в которых есть повторяющиеся слова (вне зависимости от регистра), т.е. экспериментировать нужно только с теми заклинаниями, все слова которых различны.

Напишите функцию `remove_inappropriate(spells)`, принимающую единственный параметр — список заклинаний. Функция должна удалять из этого списка особо опасные заклинания, сохраняя остальные в исходном порядке.

Функция принимает один аргумент. Им гарантированно является список строк-заклинаний. Заклинания записываются латиницей верхнего и нижнего регистра, заклинание может состоять из нескольких слов, разделённых пробелами.

Функция должна требуемым образом изменить тот самый список, который принимает в качестве параметра. Распечатка или возврат нового списка не решают поставленную задачу.

### Пример 1

Ввод

```
new_spells = ['E pluribus unum', 'Dura lex sed lex', 'dum spiro spero']
remove_inappropriate(new_spells)
print(new_spells)
```

Вывод

```
['E pluribus unum', 'dum spiro spero']
```

### Пример 2

Ввод

```
new_spells = ['Ex nihilo nihil fit', 'Nihil habenti nihil deest', 'Lorem ipsum',
              'Post hoc non est propter hoc', 'Ferro Et igni Igni et ferro', 'sum
              summus mus']
remove_inappropriate(new_spells)
print(new_spells)
```

Вывод

```
['Ex nihilo nihil fit', 'Lorem ipsum', 'sum
summus mus']
```

2. Гордей увлекается статистикой и очень хочет автоматизировать обработку данных. Разработайте функцию, которая поможет Гордею анализировать даты и находить ту из них, которая соответствует самому раннему (далёкому от сегодняшнего) дню среди весенних и осенних дней.

Функция `date_analysis(dates)` должна принимать единственный параметр — множество строк. Гордей живёт в России, поэтому каждый элемент множества представляет собой дату в формате DD.MM.YYYY (день от 01 до 31, номер месяца от 01 до 12, год от 1900 до 2019), например, двадцать шестое января тысяча девятьсот девяносто третьего года передаётся как `'26.01.1993'`. Гарантируется, что хотя бы одна осенняя или весенняя дата в наборе есть.

Функция должна принимать один параметр. Гарантируется, что им будет множество корректных дат — строк вида "DD.MM.YYYY".

Функция должна возвращать одну искомую дату в том же формате, в каком она была задана в исходном множестве. При этом не следует ничего читать с клавиатуры или выводить на экран — данные конфиденциальны!

### Пример 1

Ввод  
`dates = {"10.04.2019",  
          "01.12.2018",  
          "28.02.2019",  
          "09.04.2019"}`  
`print(date_analysis(dates))`

Вывод  
09.04.2019

### Пример 2

Ввод  
`dates = {"31.05.1999",  
          "05.01.1999",  
          "01.09.1999",  
          "01.05.2000"}`  
`print(date_analysis(dates))`

Вывод  
31.05.1999

3.Василий — менеджер магазина «Перекрёсточек». Помогите ему, разработав функцию `goods_analysis`, которая среди всех участвующих в распродаже товаров находит те, за которые дают больше всего бонусных баллов. Количество бонусных баллов рассчитывается следующим образом: за каждый полный рубль, потраченный покупателем на оплату товара с учётом скидки, начисляется 1 бонусный балл.

Функция `goods_analysis` принимает произвольное число параметров-товаров, каждый из которых представляет собой словарь со следующим набором ключей и значений:

1. ключу "название" соответствует непустая строка с наименованием товара,
2. по ключу "страна" записано название страны-производителя,
3. по ключу "цена" — соответствующее целочисленное значение в рублях,
4. ключ "скидка" отвечает за величину установленной на данный товар скидки в процентах (целое число).

Также в вашу функцию передаётся ещё один необязательный параметр — функция `in_sale`, принимающая товар в формате описанного выше словаря и возвращающая `True` для тех товаров, которые участвуют в распродаже, и `False` для остальных. Если функция `in_sale` не была передана, участвующими в распродаже считаются импортные товары, т.е. те, в качестве страны-производителя которых указана не Россия, а любая другая страна (гарантируется, что "Россия" записана только так).

Функция `goods_analysis` должна возвращать три участвующих в распродаже товара, за покупку которых начисляется больше всего бонусных баллов. Гарантируется, что такие три товара всегда найдутся. Если ответов несколько, выведите любой.

## Формат ввода

Функция должна принимать не известное заранее количество товаров заданного формата, а также необязательный именованный аргумент `in_sale`.

## Формат вывода

Возвращать функция должна три искомых товара в произвольном порядке.

### Пример 1

Ввод

```
p1, p2, p3 = goods_analysis(
    {'название': 'Колбаса конская сыровяленая', 'страна':
'Казахстан', 'цена': 240, 'скидка': 10},
    {'название': 'Копчёный лосось', 'страна': 'Норвегия', 'цена': 450,
'скидка': 5},
    {'название': 'Сыр сливочный 37%', 'страна': 'Белоруссия', 'цена':
550, 'скидка': 20},
    {'название': 'Шоколад молочный с изюмом', 'страна': 'Россия',
'цена': 65, 'скидка': 87},
)
```

```
print(*sorted([p1, p2, p3], key=lambda x: x['название']), sep='\n')
```

Вывод

```
{'название': 'Колбаса конская сыровяленая', 'страна':
'Казахстан', 'цена': 240, 'скидка': 10}
{'название': 'Копчёный лосось', 'страна': 'Норвегия', 'цена':
450, 'скидка': 5}
{'название': 'Сыр сливочный 37%', 'страна': 'Белоруссия',
'цена': 550, 'скидка': 20}
```

### Ввод

```
p1, p2, p3 = goods_analysis(  
    {'название': 'Йогурт Милая Долина 2,5%', 'страна': 'Россия',  
     'цена': 50, 'скидка': 7},  
    {'название': 'Кофеерия Капучино капсулы 24 шт.', 'страна':  
     'Россия', 'цена': 405, 'скидка': 3},  
    {'название': 'Творог 3% Находчивый Молочник', 'страна':  
     'Россия', 'цена': 50, 'скидка': 1},  
    {'название': 'Айран 0,5л', 'страна': 'Казахстан', 'цена': 61, 'скидка':  
     3},  
    {'название': 'Чёрный шоколад 95% Премиум', 'страна': 'Россия',  
     'цена': 67, 'скидка': 29},  
    {'название': 'Сметана 30% Находчивый молочник', 'страна':  
     'Россия', 'цена': 50, 'скидка': 9},  
    in_sale=lambda x: '%' in x['название']  
)  
print(*sorted([p1, p2, p3], key=lambda x: x['название']), sep='\n')
```

### Ввод

```
p1, p2, p3 = goods_analysis(  
    {'название': 'Колбаса конская сыровяленая', 'страна': 'Казахстан',  
     'цена': 240, 'скидка': 10},  
    {'название': 'Сыр сливочный 37%', 'страна': 'Белоруссия', 'цена':  
     550, 'скидка': 60},  
    {'название': 'Крекеры солёные', 'страна': 'Италия', 'цена': 250,  
     'скидка': 12},  
    {'название': 'Шоколад молочный с изюмом', 'страна': 'Россия',  
     'цена': 65, 'скидка': 87},  
    {'название': 'Копчёный лосось', 'страна': 'Норвегия', 'цена': 480,  
     'скидка': 54}  
)  
print(*sorted([p1, p2, p3], key=lambda x: x['название']), sep='\n')
```

### Вывод

```
{'название': 'Йогурт Милая Долина 2,5%', 'страна': 'Россия',  
 'цена': 50, 'скидка': 7}  
{'название': 'Творог 3% Находчивый Молочник', 'страна':  
 'Россия', 'цена': 50, 'скидка': 1}  
{'название': 'Чёрный шоколад 95% Премиум', 'страна':  
 'Россия', 'цена': 67, 'скидка': 29}
```

### Вывод

```
{'название': 'Копчёный лосось', 'страна': 'Норвегия', 'цена':  
 480, 'скидка': 54}  
{'название': 'Крекеры солёные', 'страна': 'Италия', 'цена':  
 250, 'скидка': 12}  
{'название': 'Сыр сливочный 37%', 'страна': 'Белоруссия',  
 'цена': 550, 'скидка': 60}
```