

1. Выбрать из домашней директории пользователя ubuntu файлы с расширением .py, название которых начинается на букву t.

Команда будет: `ll | grep '\<l.*\.py$'`

Демонстрация:

У меня в рабочей директории нет файлов начинающихся на "t", поэтому я буду выбирать файлы на "l"

Содержимое директории

```
→ src git:(Lesson_8) x ll
total 80
drwxr-xr-x  13 igor  staff  416B 17 ноя 17:24 __pycache__
-rw-r--r--   1 igor  staff  977B 21 ноя 21:50 deikstra.py
-rw-r--r--   1 igor  staff  2,7K 22 ноя 11:31 dfs.py
-rw-r--r--   1 igor  staff  2,0K 18 ноя 22:06 graph.py
-rw-r--r--   1 igor  staff  1,9K 22 ноя 15:45 les_8_htask_2.py
-rw-r--r--   1 igor  staff  2,7K 22 ноя 11:32 les_8_htask_3.py
-rw-rw-r--@  1 igor  staff  1,9K 17 ноя 17:47 task02.py
-rw-r--r--@  1 igor  staff  1,7K 21 ноя 22:47 task_1.py
-rw-r--r--   1 igor  staff  1,5K 25 ноя 17:30 task_2.py
-rw-r--r--@  1 igor  staff  2,9K 21 ноя 16:27 task_3.py
-rw-r--r--   1 igor  staff  1,2K 21 ноя 21:26 wide_find.py
```

Выбрать все, которые начинаются на l и оканчиваются на .py

```
→ src git:(Lesson_8) x ll | grep '\<l.*\.py$'
-rw-r--r--   1 igor  staff  1,9K 22 ноя 15:45 les_8_htask_2.py
-rw-r--r--   1 igor  staff  2,7K 22 ноя 11:32 les_8_htask_3.py
```

OK - готово

2. Из всех файлов с расширением .py, расположенных в домашней директории пользователя ubuntu, выбрать строки, содержащие команду print, и вывести их на экран.

Команда будет:

`cd ~`

`grep -H 'print' *.py`

Демонстрация:

```

→ src git:(Lesson_8) x grep -H 'print' *.py
deijkstra.py:print(dijkstra(g, s))
dfs.py:def print_graph(graph: dict):
dfs.py:    print(f'{item}: {graph[item]}')
dfs.py:    print_graph(g)
dfs.py:    print('*' * 30)
dfs.py:    print(f'for node \'{item}\': {con_component}')
graph.py:print(*graph, sep='\n')
graph.py:print('*' * 50)
graph.py:print('Ориентированный граф')
graph.py:print(*graph, sep='\n')
graph.py:print('*' * 50)
graph.py:print('Ориентированный взвешенный граф')
graph.py:print(*graph, sep='\n')
graph.py:print('*' * 50)
graph.py:print('списки смежности')
graph.py:print(*graph, sep='\n')
graph.py:print('*' * 50)
graph.py:print('Хранилище в словаре')
graph.py:print(graph_2)
graph.py:print('*' * 50)
graph.py:print('Хранилище в словаре взвешенного графа')
graph.py:print(*graph_3, sep='\n')
graph.py:    print(f'Путь из вершины 1 в 3 возможен')
graph.py:print('*' * 50)
graph.py:print('Хранилище графа в классе')
graph.py:print('*' * 50)

```

Вывод довольно объемный – поэтому обрезал его.

3. Из результатов работы команды `uptime` выведите число дней, которое система работает без перезагрузки.

Команда будет: `uptime | awk '{print $3}'`

Демонстрация:

```

→ src git:(Lesson_8) x uptime
12:25 up 6 days, 14:15, 3 users, load averages: 2,17 3,97 7,15
→ src git:(Lesson_8) x uptime | awk '{print $3}'
6

```