**Задание для УП ПМ 02.**

**Для выполнения заданий создайте глобальный репозиторий tasks-strings**

**Так же создайте папку на ПК с аналогичным именем. Все задачи необходимо выполнять в данной папке, после его запушить (залить) всё её содержимое в созданную репозиторию.**

**Задача 5.1**

Изучите программный код задач 5.1.

Для задачи 5.1 внесите в словарь данные ещё об одном устройстве.

Создайте новый файл task\_5\_1a, который должен содержать такую-же структуру, как и данная задача, только информация в словаре должна храниться, например, о характеристиках ПК.

Пример: “pc1”:{ “os”:”Windows 10”,

“Processor”:”ADM Phenom II”,

“ram”:”8 Gb”

и т.д. }

**Решение:**

computers = {  
 **"pc1"**:{  
 **"os"**:**"Windows 10"**,  
 **"Processor"**:**"ADM Phenom II"**,  
 **"ram"**:**"8 Gb"**,  
 **"motherboard"**: **"MSI87343"**,  
 **"hdd"**: **"1Tb"**,  
 },  
 **"pc2"**: {  
 **"os"**: **"Windows 8.1"**,  
 **"Processor"**: **"ADM Phenom II"**,  
 **"ram"**: **"8 Gb"**,  
 **"motherboard"**: **"ASrock-7898"**,  
 **"hdd"**: **"512gb"**,  
 },  
 **"pc3"**: {  
 **"os"**: **"Windows 7"**,  
 **"Processor"**: **"ADM Phenom II"**,  
 **"ram"**: **"4 Gb"**,  
 **"motherboard"**: **"Asus-4565"**,  
 **"hdd"**: **"1Tb"**,  
 },  
}  
  
device = input(**"Введите имя устройства: "**)  
parametr = input(**"Введите имя параметра: "**)  
print(computers[device][parametr])

**Пример:**

Введите имя устройства: pc3

Введите имя параметра: os

Windows 7

**Задача 5.2**

Изучите программный код задачи 5.2. Приведите минимум три примера разных входных данных.

**Решение:**

network = input(**"Введите адрес сети: "**)  
  
ip, mask = network.split(**"/"**)  
ip\_list = ip.split(**"."**)  
mask = int(mask)  
  
oct1, oct2, oct3, oct4 = [  
 int(ip\_list[0]),  
 int(ip\_list[1]),  
 int(ip\_list[2]),  
 int(ip\_list[3]),  
]  
  
bin\_mask = **"1"** \* mask + **"0"** \* (32 - mask)  
m1, m2, m3, m4 = [  
 int(bin\_mask[0:8], 2),  
 int(bin\_mask[8:16], 2),  
 int(bin\_mask[16:24], 2),  
 int(bin\_mask[24:32], 2),  
]  
  
ip\_output = **"""  
Network:  
{0:<8} {1:<8} {2:<8} {3:<8}  
{0:08b} {1:08b} {2:08b} {3:08b}"""**mask\_output = **"""  
Mask:  
/{0}  
{1:<8} {2:<8} {3:<8} {4:<8}  
{1:08b} {2:08b} {3:08b} {4:08b}  
"""**print(ip\_output.format(oct1, oct2, oct3, oct4))  
print(mask\_output.format(mask, m1, m2, m3, m4))

**Пример 1:**

**Введите адрес сети: 10.1.1.0/24**

**Network:**

**10 1 1 0**

**00001010 00000001 00000001 00000000**

**Mask:**

**/24**

**255 255 255 0**

**11111111 11111111 11111111 00000000**

**Пример 2:**

**Введите адрес сети: 192.168.10.2/24**

**Network:**

**192 168 10 2**

**11000000 10101000 00001010 00000010**

**Mask:**

**/24**

**255 255 255 0**

**11111111 11111111 11111111 00000000**

**Пример 3:**

**Введите адрес сети: 10.115.20.2/24**

**Network:**

**10 115 20 2**

**00001010 01110011 00010100 00000010**

**Mask:**

**/24**

**255 255 255 0**

**11111111 11111111 11111111 00000000**

**Задача 5.3**

Изучите программный код задачи 5.3. На основе данной задачи создайте новый файл 5.3a.

Внесите следующие изменения: в зависимости от того какой тип и номер интерфейса ввел пользователь необходимо выводить информацию в следующем виде:

*Fa0/1 Sector-1 connected trunk a-full a-100 10/100BaseTX  
Fa0/2 Sector-2 connected trunk a-full a-100 10/100BaseTX  
Fa0/3 Sector-3 connected trunk a-full a-100 10/100BaseTX  
Fa0/4 notconnect 1 auto auto 10/100BaseTX  
Fa0/5 port connected 100 a-full a-100 10/100BaseTX  
Fa0/6 connected trunk full 100 10/100BaseTX  
Fa0/7 disabled 100 auto auto 10/100BaseTX*

Т.е. нужно переписать строчку: print(**f"interface** {interface}**"**)

П.с. можно создать словарь с типами представленных интерфейсов и их описанием.

**Решение:**

access\_template = [  
 **"switchport mode access"**,  
 **"switchport access vlan {}"**,  
 **"switchport nonegotiate"**,  
 **"spanning-tree portfast"**,  
 **"spanning-tree bpduguard enable"**,  
]  
  
trunk\_template = [  
 **"switchport trunk encapsulation dot1q"**,  
 **"switchport mode trunk"**,  
 **"switchport trunk allowed vlan {}"**,  
]  
interface = {  
**"Fa0/1"**: **"Sector-1 connected trunk a-full a-100 10/100BaseTX"**,  
**"Fa0/2"**: **"Sector-2 connected trunk a-full a-100 10/100BaseTX"**,  
**"Fa0/3"**: **"Sector-3 connected trunk a-full a-100 10/100BaseTX"**,  
**"Fa0/4"**: **"notconnect 1 auto auto 10/100BaseTX"**,  
**"Fa0/5"**: **"port connected 100 a-full a-100 10/100BaseTX"**,  
**"Fa0/6"**: **"connected trunk full 100 10/100BaseTX"**,  
**"Fa0/7"**: **"disabled 100 auto auto 10/100BaseTX"**,  
}  
  
template = {**"access"**: access\_template, **"trunk"**: trunk\_template}  
  
mode = input(**"Введите режим работы интерфейса (access/trunk): "**)  
it = input(**"Введите тип и номер интерфейса: "**)  
vlans = input(**"Введите номер влан(ов): "**)  
  
print(interface[it])  
print(**"\n"**.join(template[mode]).format(vlans))

**Пример:**

Введите режим работы интерфейса (access/trunk): trunk

Введите тип и номер интерфейса: Fa0/1

Введите номер влан(ов): 2

Sector-1 connected trunk a-full a-100 10/100BaseTX

switchport trunk encapsulation dot1q

switchport mode trunk

switchport trunk allowed vlan 2

**Задача 5.4**

На основе таблицы (см. ниже) маршрутизаторов напишите скрипт, который бы хранил представленную информацию. При вводе названия модели необходимо выводить её характеристики.

Возьмите за основу: model = {“TE100-S5”: {“product Title”:”S-Port 10/100Mbps Fast Ethernet Switch”,

“10/100Mbps”:”5x”, и т.д.}}

Попробуйте доработать данную задачу что бы можно было вывести данные о всех устройствах у которых MAC = 1k или у кого параметр 10/100Mbps = 5x.

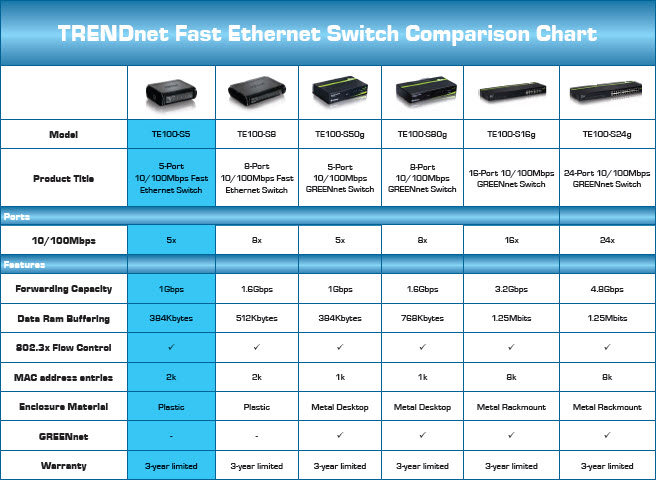


Рис. 1. Таблица моделей маршрутизаторов

**Решение:**

model = {**"TE100-S5"**:  
 {**"product Title"**:**"S-Port 10/100Mbps Fast Ethernet Switch"**,  
 **"10/100Mbps"**:**"5x"**,  
 **"Forearding Capacity"**: **"1Gps"**,  
 **"MAC adress entries"**: **"2k"**,  
 **"Enclozure Material"**: **"Plastic"**},  
 **"TE100-S8"**:  
 {**"product Title"**:**"S-Port 10/100Mbps Fast Ethernet Switch"**,  
 **"10/100Mbps"**:**"8x"**,  
 **"Forearding Capacity"**: **"1.6Gps"**,  
 **"MAC adress entries"**: **"2k"**,  
 **"Enclozure Material"**: **"Plastic"**},  
 **"TE100-S50g"**:  
 {**"product Title"**:**"S-Port 10/100Mbps GREENnet Switch"**,  
 **"10/100Mbps"**:**"5x"**,  
 **"Forearding Capacity"**: **"1Gps"**,  
 **"MAC adress entries"**: **"1k"**,  
 **"Enclozure Material"**: **"Metal Desktop"**},  
 **"TE100-S80g"**:  
 {**"product Title"**:**"S-Port 10/100Mbps GREENnet Switch"**,  
 **"10/100Mbps"**:**"8x"**,  
 **"Forearding Capacity"**: **"1.6Gps"**,  
 **"MAC adress entries"**: **"1k"**,  
 **"Enclozure Material"**: **"Metal Desktop"**},  
 **"TE100-S16g"**:  
 {**"product Title"**:**"S-Port 10/100Mbps GREENnet Switch"**,  
 **"10/100Mbps"**:**"16x"**,  
 **"Forearding Capacity"**: **"3.2Gps"**,  
 **"MAC adress entries"**: **"8k"**,  
 **"Enclozure Material"**: **"Metal Rackmount"**},  
 **"TE100-S24g"**:  
 {**"product Title"**:**"S-Port 10/100Mbps GREENnet Switch"**,  
 **"10/100Mbps"**:**"24x"**,  
 **"Forearding Capacity"**: **"4.8Gps"**,  
 **"MAC adress entries"**: **"8k"**,  
 **"Enclozure Material"**: **"Metal Rackmount"**},  
 }  
  
**for** switch **in** model.keys():  
 **if** model[switch][**'10/100Mbps'**] == **"5x" and** \  
 model[switch][**'MAC adress entries'**] == **"1k"**:  
 print(model[switch])

**Пример:**

{'product Title': 'S-Port 10/100Mbps GREENnet Switch', '10/100Mbps': '5x', 'Forearding Capacity': '1Gps', 'MAC adress entries': '1k', 'Enclozure Material': 'Metal Desktop'}

**Задача 5.5**

На основе предыдущей задачи вынесите данные об устройствах во внешний файл json. Напишите программу, которая работает с этим файлом.

**Функционал:**

* Открывает
* Выводит данные
* Выводит данные по параметрам
* Сохраняет изменения

Руководство по работе с json из python:

<https://pyneng.readthedocs.io/ru/latest/book/17_serialization/json.html>