**Тема - Циклы**

*Задачи на использование словарей*

**Задание 1:** Запросить у пользователя ввод IP-адреса в формате 10.0.1.1. Ввод данных осуществляется в виде строки. В зависимости от типа адреса (описаны ниже), вывести на стандартный поток вывода:

• «unicast» - если первый байт в диапазоне 1-223

• «multicast» - если первый байт в диапазоне 224-239

• «local broadcast» - если IP-адрес равен 255.255.255.255

• «unassigned» - если IP-адрес равен 0.0.0.0

• «unused» - во всех остальных случаях

Ограничение: Все задания надо выполнять используя только пройденные темы.

**Задание 2:** Сделать копию скрипта задания 1. Добавить проверку введенного IP-адреса. Адрес считается корректно заданным, если он:

• состоит из 4 чисел (а не букв или других символов)

• числа разделенны точкой

• каждое число в диапазоне от 0 до 255

Если адрес задан неправильно, выводить сообщение: «Неправильный IP-адрес». Сообщение «Неправильный IP-адрес» должно выводиться только один раз, даже если несколько пунктов выше не выполнены.

Ограничение: Все задания надо выполнять используя только пройденные темы.

**Задание 3**: В скрипте сделан генератор конфигурации для access-портов. Сделать аналогичный генератор конфигурации для портов trunk. В транках ситуация усложняется тем, что VLANов может быть много, и надо понимать, что с ним делать. Поэтому в соответствии каждому порту стоит список и первый (нулевой) элемент списка указывает как воспринимать номера VLAN, которые идут дальше.

Пример значения и соответствующей команды:

• [«add», «10», «20»] - команда switchport trunk allowed vlan add 10,20

• [«del», «17»] - команда switchport trunk allowed vlan remove 17

• [«only», «11», «30»] - команда switchport trunk allowed vlan 11,30

Задача для портов 0/1, 0/2, 0/4:

• сгенерировать конфигурацию на основе шаблона trunk\_template

• с учетом ключевых слов add, del, only

Код не должен привязываться к конкретным номерам портов. То есть, если в словаре trunk будут другие номера интерфейсов, код должен работать.

Для данных в словаре trunk\_template вывод на стандартный поток вывода должен быть таким

interface FastEthernet 0/1

switchport trunk encapsulation dot1q

switchport mode trunk

switchport trunk allowed vlan add 10,20

interface FastEthernet 0/2

switchport trunk encapsulation dot1q

switchport mode trunk

switchport trunk allowed vlan 11,30

interface FastEthernet 0/4

switchport trunk encapsulation dot1q

switchport mode trunk

switchport trunk allowed vlan remove 17

Генератор для access портов представлен ниже

access\_template = [

"switchport mode access",

"switchport access vlan",

"spanning-tree portfast",

"spanning-tree bpduguard enable",

]

trunk\_template = [

"switchport trunk encapsulation dot1q",

"switchport mode trunk",

"switchport trunk allowed vlan",

]

access = {"0/12": "10", "0/14": "11", "0/16": "17", "0/17": "150"}

trunk = {"0/1": ["add", "10", "20"], "0/2": ["only", "11", "30"], "0/4": ["del", "17"]}

for intf, vlan in access.items():

print("interface FastEthernet" + intf)

for command in access\_template:

if command.endswith("access vlan"):

print(f" {command} {vlan}")

else:

print(f" {command}")

**Задание 4**: Написать программу – игра - угадай, в которой с помощью функции random генерируется случайное число от 1 до 50. Пользователю предлагается угадать данное число, на основе подсказать – загаданное число больше или меньше числа пользователя. Вывести число попыток отгадывания числа

**Задание 5**: Написать программу. Пользователь вводит два числа. Данные числа определяют числовой диапазон, для которого надо найти все числа, которые делятся нацело на 3, 5, 9. Например диапазон 1 - 20, количество чисел, которые делятся на 3 – 6, 5 – 4, 9 - 2. Важно учесть, что пользователь может ввести числа в обратном порядке, например 37 и 16.