

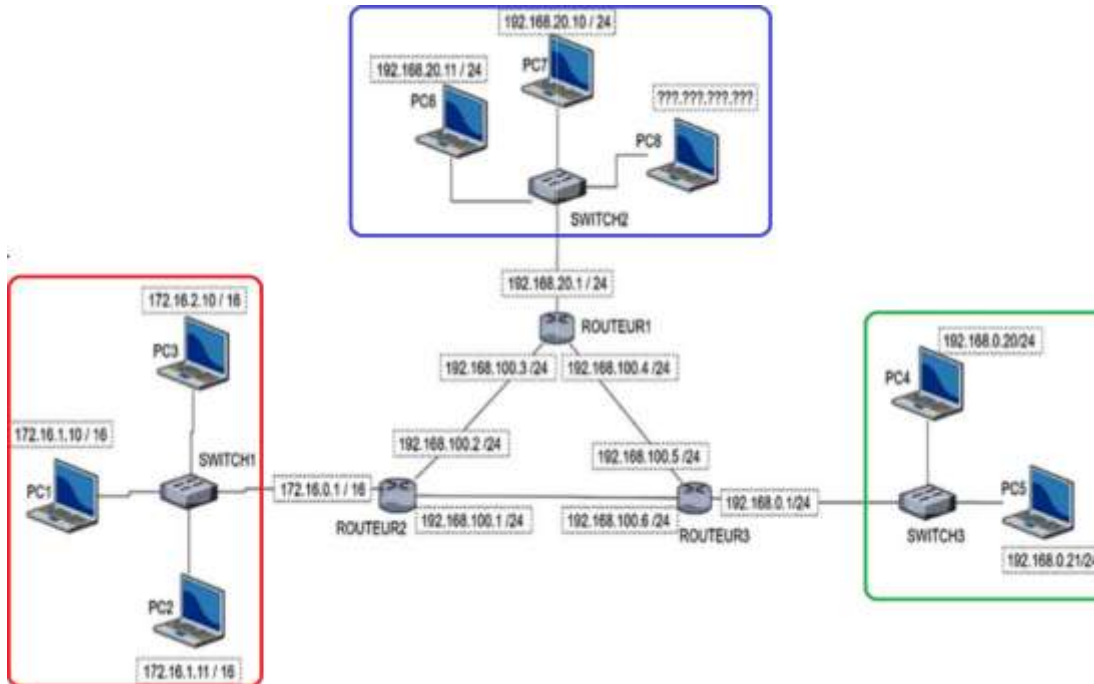
Correction

NSI - 2021 Étranger Jour 1 (21-NSIJ1G11)

Exercice 4 - Étude d'un réseau informatique

PARTIE A : Étude de l'adressage IP

1.



2. a. Quatre octets sont nécessaire pour composer une adresse IPv4.

2. b. $168_{(10)} = 10101000_{(2)}$ et $10_{(10)} = 00001010_{(2)}$

Adresse IP (V4) du PC7	Ligne 1	192	168	20	10
	Ligne 2	1 1 0 0 0 0 0 0	10101000	0 0 0 1 0 1 0 0	00001010

2. c. et d.

Masque de sous réseau	Ligne 3	11111111	11111111	11111111	00000000
	Ligne 4	255	255	255	0

i. et ii.

Pour obtenir l'adresse réseau binaire, on réalise un ET(&) logique entre chaque bit de l'adresse IP (ligne 2) et du masque de sous réseau (ligne3)					
Adresse du réseau	Ligne 5	11000000	10101000	00010100	00000000
	Ligne 6	192	168	20	0

3. Pour connecter le PC8, les adresses IP possibles sont :

- ~~192.168.20.0~~ Réservé pour l'adresse du sous-réseau
- ~~192.256.20.14~~ Mauvaise adresse du sous-réseau. Quand bien même, il y aurait un conflit d'adresse IP avec le PC6.
- 192.168.20.30
- 192.168.20.230
- ~~192.168.20.260~~ Adresse impossible. La valeur maximale est 254. Le 255 est réservé pour le broadcast.
- ~~192.168.27.14~~ L'adresse du sous-réseau n'est pas correcte

Correction

NSI - 2021 Étranger Jour 1 (21-NSIJ1G11)

PARTIE B : Une fonction pour convertir une adresse IP en décimal pointé en notation binaire.

```
def IP_bin(add_IP):  
    """ Renvoie la conversion en binaire d'une adresse IP décimale. """  
    add_IP_bin = []  
    for n in add_IP:  
        add_IP_bin.append(dec_bin(n))  
    return add_IP_bin
```

Hors-sujet :

```
def dec_bin(n):  
    """ Convertie l'entier n en binaire (liste). """  
    n = list(str(bin(n))[2:])  
    n = list(map(lambda x: int(x), n))  
    while len(n) < 8:  
        n = [0]+n  
    return n
```