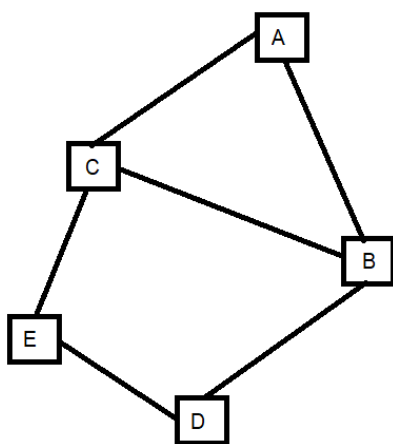


Proposition de correction

Exercice 1

Partie A

Q1



Q2

Routeur E		
Destination	Passe par	Distance
A	C	2
B	D	2
C	C	1
D	D	1

Q3

Routeur A		
Destination	Passe par	Distance
B	B	1
C	B	2
D	B	2
E	C	2

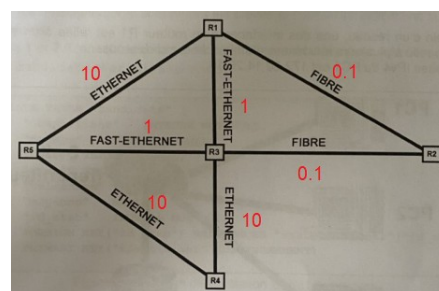
Partie B

Q1

Type de Liaison	Ethernet	Fast-Ethernet	Fibre
Débit	10^7	10^8	10^9
Coût	10	1	0,1

Q2, Q2, Q3

$R1 \rightarrow R2 \rightarrow R3 \rightarrow R4$	$c = 0,1 + 0,1 + 10 = 10,2$
$R1 \rightarrow R2 \rightarrow R3 \rightarrow R5 \rightarrow R4$	$c = 0,1 + 0,1 + 1 + 10 = 11,2$
$R1 \rightarrow R3 \rightarrow R4$	$c = 1 + 10 = 11$
$R1 \rightarrow R3 \rightarrow R5 \rightarrow R4$	$c = 1 + 1 + 10 = 12$
$R1 \rightarrow R5 \rightarrow R4$	$c = 10 + 10 = 20$
$R1 \rightarrow R5 \rightarrow R3 \rightarrow R4$	$c = 10 + 1 + 10 = 21$



Partie C

Q1

$172.20.14.26/16 \rightarrow @réseau = 172.20.0.0/16$

Q2

$256 - 2 - 1$ (routeur) = 253 machines

Q3

172.20.14.3

Q4

10101100.00010100.00001110.00010000

Q5

ifconfig (Unix), ipconfig (MS-Windows)

Exercice 2

Q1.a

candidats (identifiant : INT, prenom : TEXT, nom : TEXT, email : TEXT, telephone : TEXT, moyenne : REAL, #id_etab : INT)

etablisements (id_etab : INT, etablisement : TEXT, adresse : TEXT, email : TEXT, proviseur : TEXT)

Q1.b

id_etab : clé primaire de la relation etablisements

Q1.c

INSERT INTO candidats

VALUES(8, 'Jacques', 'HADDY', 'jacques.haddy@gmail.com', '98 76 54 32 10', 9.75, 2)

Q2.a

prenom	nom
Lydie	HECTOR
Mathis	PARIS
Roman	LAMI

Q2.b

```
SELECT candidats.nom, candidats.prenom, candidats.moyenne
FROM candidats, etablisements
WHERE etablisements.id_etab = 1
AND candidats.id_etab = etablisements.id_etab
ORDER BY candidats.nom, candidats.prenom
```

Q2.c

nom	prenom
APPA	Kevin
PASTA	Pierre
TARDI	Paul
TOU	Claire

Q3

```
SELECT etablisements.etablissement, etablisements.proviseur
FROM etablisements, candidats
WHERE candidats.id_etab = etablisements.id_etab
ORDER BY candidats.moyenne ASC
```

Exercice 3**Q1.a**

$4 - 1 + 3 - 2 = 4$

Q1.b

prendre le maximum des 2 cases suivantes

Q2.a

```
def parcours1(e : list) -> int :
    # return sum(e)
    somme = 0
    for case in e :
        somme += case
    return somme
```

Q2.b

```
def parcours2(e : list) -> int :
    somme = 0
    for case in range(0, len(e), 2) :
        somme += case
    return somme
```

Q3

```
def score(plateau : list, déplacements : str) -> int :
    somme = 0
    i = 0
    while i < len(deplacements) :
        if deplacements[i] == 'D' :
            i += 1
        else :
            i += 2
        somme += plateau[i]
    return somme
```

Q4

```
def meilleur(plateau):
    score_max = parcour1(plateau)
    for i in range(1000) :
        d = genere(len(plateau))
        s = score(plateau, d)
        if score_max < s:
            score_max = s
    return score_max
```

Q5

```
def genere(n : int) -> str :
    déplacements = ""
    i = 0
    while i < n :
        if randint(0,1) and (i < n - 1) :
            déplacements += 'XS'
            i += 2
        else :
            déplacements += 'D'
            i += 1
    return déplacements
```