Bac Général Maths + NSI (Mathématiques et Numérique et sciences informatiques)

SUJET D'EXAMEN (SESSION OFFICIELLE)

Numérique et sciences informatiques - Session 2023

Sujet de l'examen national tel que publié pour la session indiquée. Source : sujets nationaux publiés par le ministère chargé de l'examen et les services académiques. Document mis à disposition par FormaV pour la révision.

Toutes les annales et leurs corrigés : https://bac-maths-nsi.fr/annales-sujets-et-corriges

BACCALAURÉAT GÉNÉRAL

ÉPREUVE D'ENSEIGNEMENT DE SPÉCIALITÉ

SESSION 2023

NUMÉRIQUE ET SCIENCES INFORMATIQUES

JOUR 2

Durée de l'épreuve : **3 heures 30**

L'usage de la calculatrice n'est pas autorisé.

Dès que ce sujet vous est remis, assurez-vous qu'il est complet.
Ce sujet comporte 8 pages numérotées de 1/8 à 8/8.

Le candidat traite les 3 exercices proposés.

EXERCICE 1 (4 points)

Cet exercice comporte trois parties indépendantes.
Il porte sur la notion de protocoles réseaux, sur l'adressage IP et les systèmes d'exploitation.

Partie A

Le protocole RIP (Routing Information Protocol) est un protocole de routage qui cherche à minimiser le nombre de routeurs traversés (ce qui correspond à la distance ou au nombre de sauts).
On considère un réseau informatique composé de cinq routeurs (A, B, C, D et E), chacun associé directement à un réseau du même nom obtenu avec le protocole RIP.

1. On donne ci-dessous les tables de routage des routeurs A, B, C et D.
Schématiser, en vous aidant de ces tables de routage, le réseau composé par les cinq routeurs.

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

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

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

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

Tables de routage

2. En déduire une table de routage possible pour le routeur E.
3. La connexion entre le routeur A et C est coupée. Déterminer alors la nouvelle table de routage du routeur A.

Partie B

Le protocole OSPF (Open Shortest Path First) est un protocole servant à déterminer le chemin le plus rapide (le coût total le plus petit) que peuvent emprunter des paquets de données dans un réseau de routeurs.

Le coût d'une liaison entre deux routeurs est donné par la formule :

$$\text{coût} = \frac{10^8}{d}$$

où d correspond au débit exprimé en bit/s.

On considère un nouveau réseau (figure 1) composé de cinq routeurs (R1, R2, R3, R4 et R5) ci-dessous :

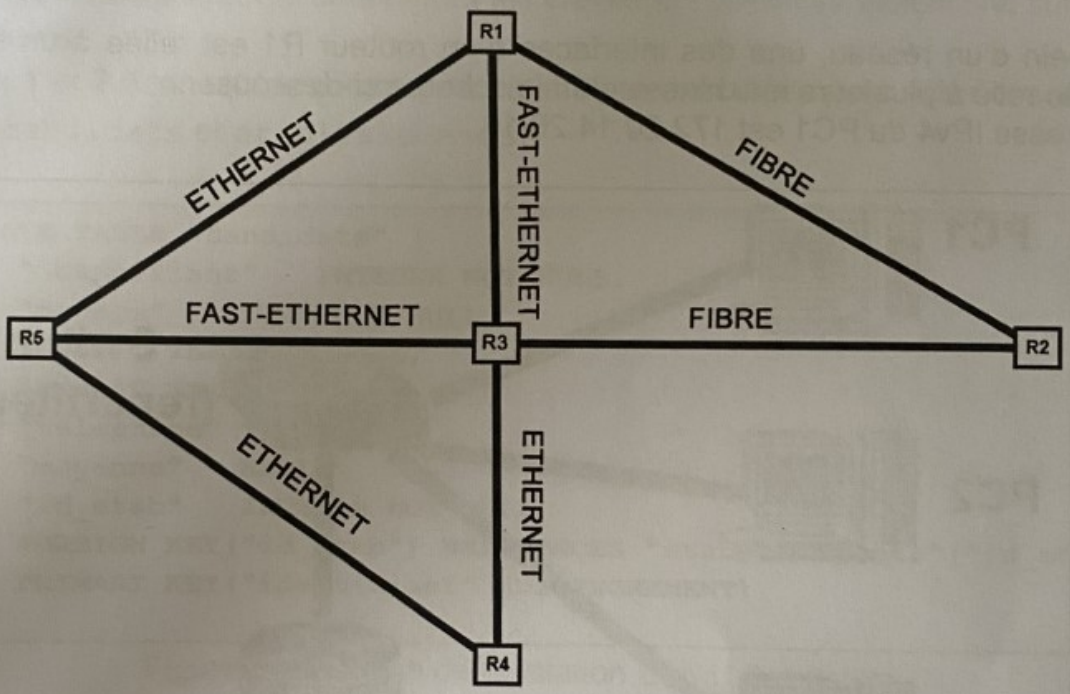


Figure 1 : schéma du réseau

1. Recopier et compléter le tableau ci-dessous :

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

On souhaite acheminer un paquet de données depuis le routeur R1 au routeur R4.

2. Donner la liste des six chemins possibles que peut emprunter ce paquet.

3. Déterminer le coût total de chacun de ces chemins puis en déduire le chemin respectant le protocole OSPF.

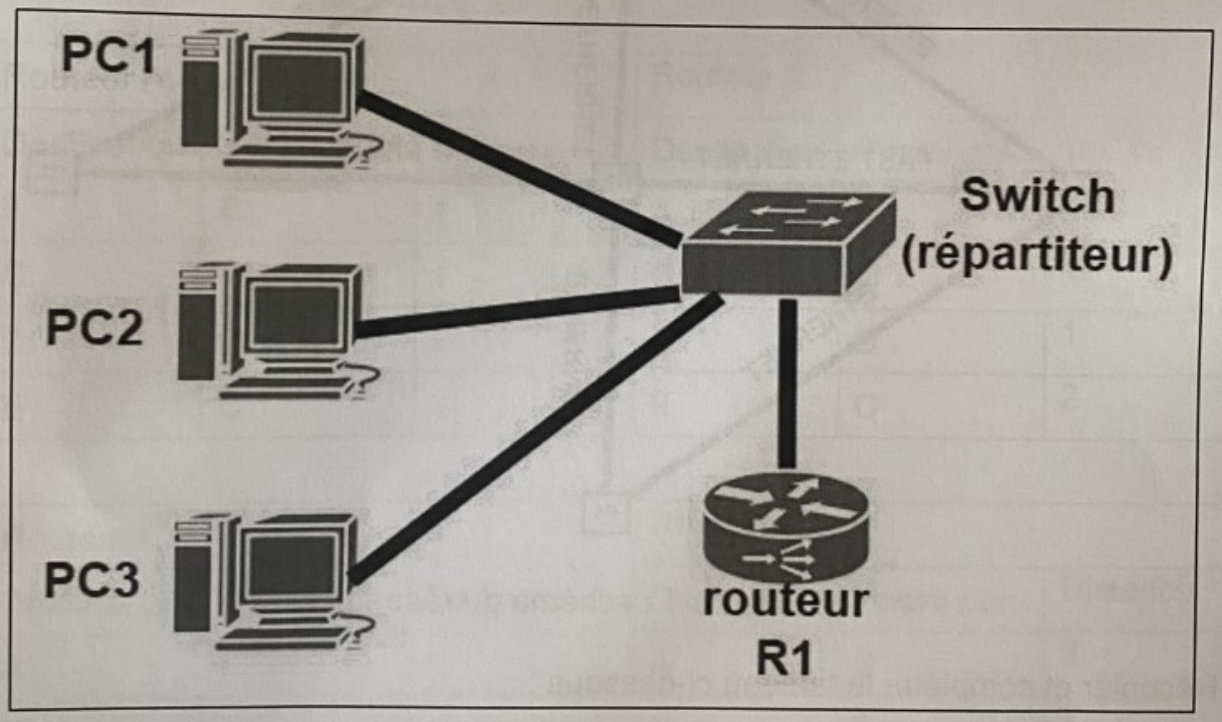
Partie C

Quelques rappels :
Les adresses IP seront conformes à la norme IPv4, à savoir composées de 4 octets. Elles prendront la forme X1.X2.X3.X4, où X1, X2, X3 et X4 sont les valeurs des 4 octets convertis en notation décimale.

La notation CIDR X1.X2.X3.X4/n signifie que les n premiers bits de poids forts de l'adresse IP représentent la partie « réseau », les bits suivants la partie « hôte » (machine).

Toutes les adresses des hôtes connectés à un réseau local ont la même partie réseau et peuvent donc communiquer directement.

Au sein d'un réseau, une des interfaces d'un routeur R1 est reliée à un switch lui-même relié à plusieurs machines selon le schéma ci-dessous.
L'adresse IPv4 du PC1 est 172.20.14.26/16.



1. Donner l'adresse de ce réseau.
2. En déduire le nombre de machines pouvant être connectées à ce réseau.
3. L'interface du routeur a pour adresse IPv4 172.20.14.254/16.
   Donner une adresse IPv4 possible pour la machine PC3 appartenant à ce réseau.
4. L'une de ces machines a pour adresse IPv4 172.20.14.16.
   Déterminer l'écriture binaire de l'adresse de cette machine.
5. L'utilisateur du PC2 souhaiterait connaître l'adresse IPv4 de sa machine.
   Donner la commande du terminal du système d'exploitation du PC2 permettant de l'obtenir.

Exercice 2 (3 points)

Cet exercice porte sur les bases de données relationnelles et le langage SQL.

Dans cet exercice, on pourra utiliser les mots clés suivants du langage SQL : `SELECT`, `FROM`, `WHERE`, `JOIN...ON`, `INSERT INTO...VALUES...`, `UPDATE...SET`, `ORDER BY`.
La clause `ORDER BY` suivie d'un attribut permet de trier les résultats par ordre croissant des valeurs de l'attribut.

On veut gérer les résultats d'un groupe d'élèves au baccalauréat à l'aide d'une base de données relationnelle en langage SQL. Pour cela, on a créé deux relations : une contenant les renseignements concernant les élèves et l'autre des indications sur les établissements.
Les figures 1 et 2 donnent les commandes ayant été exécutées pour créer ces deux relations : `candidats` et `etablissements`.

```
CREATE TABLE "candidats" (
    "identifiant"   INTEGER NOT NULL,
    "prenom"    TEXT NOT NULL,
    "nom"   TEXT NOT NULL,
    "email" TEXT,
    "telephone" TEXT,
    "moyenne"   REAL,
    "id_etab"   INTEGER NOT NULL,
    FOREIGN KEY("id_etab") REFERENCES "etablissements"("id_etab"),
    PRIMARY KEY("identifiant" AUTOINCREMENT)
);
```

Figure 1 : Création de la relation `candidats`

```
CREATE TABLE "etablissements" (
    "id_etab"   INTEGER NOT NULL,
    "etablissement" TEXT,
    "adresse"   TEXT,
    "mail"  TEXT,
    "proviseur" TEXT,
    PRIMARY KEY("id_etab" AUTOINCREMENT)
);
```

Figure 2 : Création de la relation `etablissements`.

Les figures 3 et 4 ci-dessous donnent les contenus de ces relations.

	identifiant	prenom	nom	email	telephone	moyenne	id_etab
	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre
1	1	Kevin	APPA	kevin.appa@gmail.com	0102030405	12.2	1
2	2	Paul	TARDI	paul.tardi@gmail.com	1539457825	11.5	3
3	3	Pierre	PASTA	pierre.pasta@gmail.com	1535687725	11.5	3
4	4	Claire	TOU	claire.tou@gmail.com	1535712825	9.5	1
5	5	Lydie	HECTOR	lydie.hector@gmail.com	9895712825	18.5	2
6	6	Mathis	PARIS	mathis.paris@gmail.com	9895719999	19.5	1
7	7	Roman	LAMI	roman.lami@gmail.com	9894758999	19.5	2

Figure 3 : Contenu de la relation `candidats`.

	id_etab	etablissement	adresse	mail	proviseur
	Filtre	Filtre	Filtre	Filtre	Filtre
1	1	Vinci	2 rue de la gare trelouper	vinci@rectorat.com	Gerard LECHEF
2	2	Acamus	3 rue du bord de l'eau saturne 8066	acamus@rectorat.com	Camille PREMIERE
3	3	Gbrassens	avenue de Paris 60555 Beaupre	gbrassens@rectorat.com	Philippe STAR

Figure 4 : Contenu de la relation `etablissements`.

1.

a. Représenter par un schéma relationnel les relations `candidats` et `etablissements` en précisant les types pour les attributs.

b. Indiquer, en justifiant, quelle est la clé étrangère de la relation `candidats`.

c. Écrire une requête permettant d'ajouter le candidat présentant les renseignements suivants :

- identifiant : 8
- email : jacques.haddy@gmail.com
- moyenne : 9.75
- nom : HADDY
- prenom : Jacques
- telephone : 98 76 54 32 10
- etablissement : Acamus

2.

a. Écrire ce qu'affiche la requête suivante appliquée aux relations précédentes :

```
SELECT prenom, nom
FROM candidats WHERE moyenne > 12.5 ;
```

b. Écrire une requête permettant d'afficher les noms, prénoms et moyennes des candidats venant de l'établissement dont l'`id_etab` vaut 1.

c. Écrire ce qu'affiche la requête suivante :

```
SELECT nom, moyenne
FROM candidats WHERE moyenne < 16 ORDER BY nom ;
```

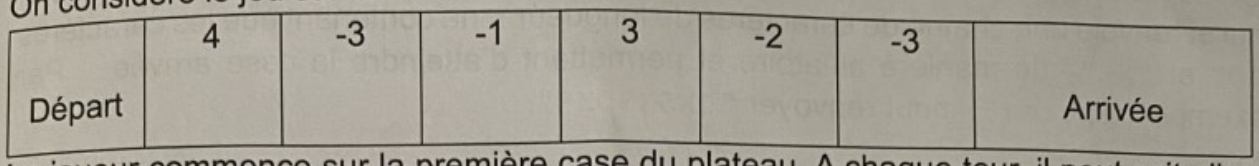
3.

Écrire une requête donnant les noms des établissements et de leurs proviseurs classés par ordre croissant des notes.

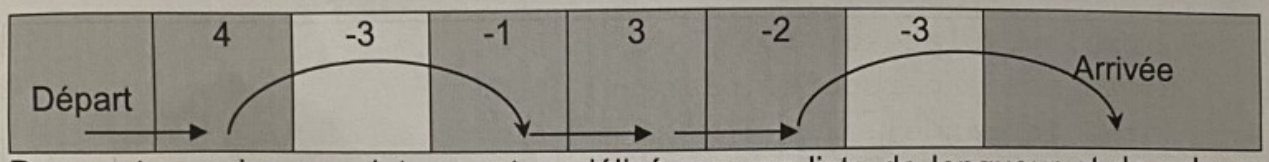
EXERCICE 3 (5 points)

Cet exercice porte sur les algorithmes.

On considère le jeu suivant :



Le joueur commence sur la première case du plateau. A chaque tour, il peut soit aller sur la case juste à droite, soit la sauter. A chaque fois qu'il passe sur une case, il gagne le nombre de points affichés sur celle-ci. L'objectif est d'atteindre la case arrivée en maximisant le nombre de points. Par exemple, pour le jeu précédent, le maximum possible est 4 en prenant le chemin grisé :



Dans cet exercice, un plateau est modélisé par une liste de longueur et de valeurs variables.

Par exemple, la liste du plateau précédent est : `[4,-3, -1, 3, -2, -3]`.

1.

a. Donner sans le justifier le plus haut score possible, si le plateau est défini par la liste `[-2, 4, -1, -2, 1, 4]`.

b. Si le plateau ne contient que des valeurs positives, quelle est la meilleure stratégie ?

2.

a. Écrire une fonction `parcours1(e)` qui prend en paramètre une liste `e` représentant le plateau et renvoie le score réalisé si le joueur choisit comme stratégie de passer par toutes les cases.

Pour la question suivante, on suppose que le nombre de cases est impair.

b. Écrire une fonction `parcours2(e)` qui prend en paramètre une liste `e` représentant le plateau et renvoie le score réalisé si le joueur choisit comme stratégie de ne faire que des sauts.

3. On modélise les choix du joueur par une chaîne de caractères qui contient les caractères `"D"` (le joueur se déplace sur la case juste à droite) ou `"XS"` (le joueur saute une case toujours sur la droite).

Écrire une fonction `score(plateau, deplacements)` qui reçoit en paramètre un plateau sous forme de liste et un choix de déplacements donné sous forme d'une chaîne de caractères et retourne le score réalisé par ce choix. Dès que l'arrivée est atteinte, le score est renvoyé. On considère que la chaîne de caractères permet d'atteindre la case arrivée.

Par exemple :
`score([4, -3, -1, 3, -2, -3], "DDXSDDD")` renvoie -1
résultat du calcul 4-3+3-2-3 =-1

On suppose que la fonction `genere(n)` reçoit en paramètre un entier naturel `n` non nul et renvoie une chaîne de caractères de longueur `n` ne contenant que les caractères `"D"` et `"XS"` de manière aléatoire et permettant d'atteindre la case arrivée. Par exemple `genere(8)` peut renvoyer `"DXSXSDXS"`.

4.

On souhaite écrire une fonction `meilleur(plateau)` qui génère 1000 déplacements dont la longueur est le nombre de cases du plateau pour s'assurer d'arriver au bout et qui renvoie le meilleur score sur ces 1000 déplacements. Recopier puis compléter la fonction ci-dessous :

```
1 def meilleur(plateau) :
2     score_max = parcours1(plateau)
3     for i in range(.....) :
4         d = genere(.....)
5         s = score(plateau, d)
6         if ..... :
7             score_max = s
8     return score_max
```

5.

Ecrire la fonction `genere(n)` de la question 3. On pourra à cet effet utiliser la fonction `randint` du module `random` qui reçoit deux entiers `a` et `b` et renvoie un entier de l'intervalle `[a,b]` avec `a` et `b` inclus.