BACCALAURÉAT

SESSION 2021

---

Épreuve de l'enseignement de spécialité

# NUMÉRIQUE et SCIENCES INFORMATIQUES

## Partie écrite

## Classe terminale de la voie générale

---

Sujet zéro

---

DURÉE DE L'ÉPREUVE : 3 heures 30 min

**Le sujet comporte 12 pages numérotées de 1/12 à 12/12.**
**Dès que le sujet vous est remis, assurez-vous qu'il est complet.**

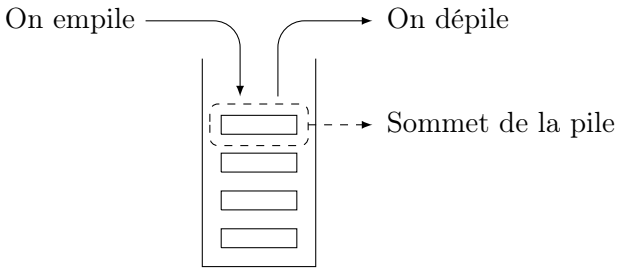
*Le candidat doit choisir 3 exercices qu'il traitera sur les 5 exercices proposés.*
*Il est rappelé que la qualité de la rédaction, la clarté et la précision des raisonnements entreront pour une part importante dans l'appréciation des copies.*

***L'usage de la calculatrice est interdit.***

# Exercice 1

*Cet exercice porte sur la notion de pile et sur la programmation de base en Python.*

On rappelle qu'une pile est une structure de données abstraite fondée sur le principe « dernier arrivé, premier sorti » :

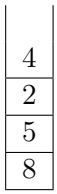



*Une pile*

On munit la structure de données Pile de quatre fonctions primitives définies dans le tableau ci-dessous. :

| **Structure de données abstraite** : Pile |
|---|
| **Utilise** : Éléments, Booléen |
| **Opérations** :<br>— `creer_pile_vide : ∅ ⟶ Pile`<br>`creer_pile_vide()` renvoie une pile vide<br>— `est_vide : Pile ⟶ Booléen`<br>`est_vide(pile)` renvoie True si `pile` est vide, False sinon<br>— `empiler : Pile, Élément ⟶ Rien`<br>`empiler(pile, element)` ajoute `element` au sommet de la `pile`<br>— `depiler : Pile ⟶ Élément`<br>`depiler(pile)` renvoie l'élément au sommet de la `pile` en le retirant de la `pile` |

**Question 1** On suppose dans cette question que le contenu de la pile `P` est le suivant (les éléments étant empilés par le haut) :




Quel sera le contenu de la pile `Q` après exécution de la suite d'instructions suivante ?

```
1       Q = creer_pile_vide()
2       while not est_vide(P):
3           empiler(Q, depiler(P))
```

**Question 2**

1. On appelle *hauteur* d'une pile le nombre d'éléments qu'elle contient. La fonction `hauteur_pile` prend en paramètre une pile `P` et renvoie sa hauteur. Après appel de cette fonction, la pile `P` doit avoir retrouvé son état d'origine.

   **Exemple :** si `P` est la pile de la question 1 : `hauteur_pile(P) = 4`.

   Recopier et compléter sur votre copie le programme Python suivant implémentant la fonction `hauteur_pile` en remplaçant les `???` par les bonnes instructions.

```
def hauteur_pile(P):
    Q = creer_pile_vide()
    n = 0
    while not(est_vide(P)):
        ???
        x = depiler(P)
        empiler(Q,x)
    while not(est_vide(Q)):
        ???
        empiler(P, x)
    return ???
```

2. Créer une fonction `max_pile` ayant pour paramètres une pile `P` et un entier `i`. Cette fonction renvoie la position `j` de l'élément maximum parmi les `i` derniers éléments empilés de la pile `P`. Après appel de cette fonction, la pile `P` devra avoir retrouvé son état d'origine. La position du sommet de la pile est `1`.

   **Exemple :** si `P` est la pile de la question 1 : `max_pile(P, 2) = 1`

**Question 3** Créer une fonction `retourner` ayant pour paramètres une pile `P` et un entier `j`. Cette fonction inverse l'ordre des `j` derniers éléments empilés et ne renvoie rien. On pourra utiliser deux piles auxiliaires.

**Exemple :** si `P` est la pile de laquestion 1(a), après l'appel de `retourner(P, 3)`, l'état de la pile `P` sera :


**Question 4** L'objectif de cette question est de trier une pile de crêpes.

On modélise une pile de crêpes par une pile d'entiers représentant le diamètre de chaque crêpe. On souhaite réordonner les crêpes de la plus grande (placée en bas de la pile) à la plus petite (placée en haut de la pile).

On dispose uniquement d'une spatule que l'on peut insérer dans la pile de crêpes de façon à retourner l'ensemble des crêpes qui lui sont au-dessus.

Le principe est le suivant :

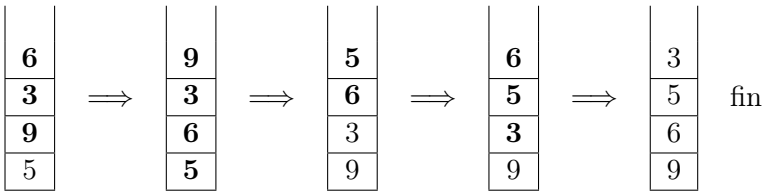
— On recherche la plus grande crêpe.

— On retourne la pile à partir de cette crêpe de façon à mettre cette plus grande crêpe tout en haut de la pile.

— On retourne l'ensemble de la pile de façon à ce que cette plus grande crêpe se retrouve tout en bas.

— La plus grande crêpe étant à sa place, on recommence le principe avec le reste de la pile.

**Exemple :**




Créer la fonction `tri_crepes` ayant pour paramètre une pile `P`. Cette fonction trie la pile `P` selon la méthode du tri crêpes et ne renvoie rien. On utilisera les fonctions créées dans les questions précédentes.

**Exemple :** Si la pile `P` est

| |
|---|
| 7 |
| 14 |
| 12 |
| 5 |
| 8 |

, après l'appel de `tri_crepes(P)`, la pile `P` devient

| |
|---|
| 5 |
| 7 |
| 8 |
| 12 |
| 14 |

.

# Exercice 2

*Cet exercice porte sur la programmation en général et la récursivité en particulier.*

On considère un tableau de nombres de $n$ lignes et $p$ colonnes.

Les lignes sont numérotées de 0 à $n-1$ et les colonnes sont numérotées de 0 à $p-1$. La case en haut à gauche est repérée par $(0,0)$ et la case en bas à droite par $(n-1, p-1)$.

On appelle *chemin* une succession de cases allant de la case $(0,0)$ à la case $(n-1, p-1)$, en n'autorisant que des déplacements case par case : soit vers la droite, soit vers le bas.

On appelle *somme* d'un chemin la somme des entiers situés sur ce chemin.

Par exemple, pour le tableau `T` suivant :

| | | | |
|---|---|---|---|
| **4** | **1** | **1** | 3 |
| 2 | 0 | **2** | 1 |
| 3 | 1 | **5** | **1** |

— Un chemin est $(0,0)$, $(0,1)$, $(0,2)$, $(1,2)$, $(2,2)$, $(2,3)$ (en gras sur le tableau) ;

— La somme du chemin précédent est 14.

— $(0,0)$, $(0,2)$, $(2,2)$, $(2,3)$ n'est pas un chemin.

L'objectif de cet exercice est de déterminer la somme maximale pour tous les chemins possibles allant de la case $(0,0)$ à la case $(n-1, p-1)$.

**Question 1** On considère tous les chemins allant de la case $(0,0)$ à la case $(2,3)$ du tableau `T` donné en exemple.

1. Un tel chemin comprend nécessairement 3 déplacements vers la droite. Combien de déplacements vers le bas comprend-il ?
2. La longueur d'un chemin est égal au nombre de cases de ce chemin. Justifier que tous les chemins allant de $(0,0)$ à $(2,3)$ ont une longueur égale à 6.

**Question 2** En listant tous les chemins possibles allant de $(0,0)$ à $(2,3)$ du tableau `T`, déterminer un chemin qui permet d'obtenir la somme maximale et la valeur de cette somme.

**Question 3** On veut créer le tableau `T'` où chaque élément `T'[i][j]` est la somme maximale pour tous les chemins possibles allant de $(0,0)$ à $(i,j)$.

1. Compléter et recopier sur votre copie le tableau `T'` donné ci-dessous associé au tableau

`T` =

| | | | |
|---|---|---|---|
| 4 | 1 | 1 | 3 |
| 2 | 0 | 2 | 1 |
| 3 | 1 | 5 | 1 |

`T'` =

| | | | |
|---|---|---|---|
| 4 | 5 | 6 | 9 |
| 6 | 6 | 8 | 10 |
| 9 | 10 | 15 | 16 |

2. Justifier que si j est différent de 0, alors : T'[0][j] = T[0][j] + T'[0][j-1]

**Question 4** Justifier que si i et j sont différents de 0, alors : T'[i][j] = T[i][j] + max(T'[i-1][j], T'[i][j-1]).

**Question 5** On veut créer la fonction récursive `somme_max` ayant pour paramètres un tableau T, un entier i et un entier j. Cette fonction renvoie la somme maximale pour tous les chemins possibles allant de la case $(0,0)$ à la case $(i,j)$.

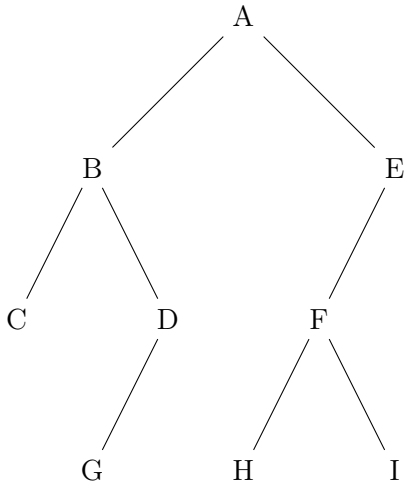
1. Quel est le cas de base, à savoir le cas qui est traité directement sans faire appel à la fonction `somme_max` ? Que renvoie-t-on dans ce cas ?
2. À l'aide de la question précédente, écrire en Python la fonction récursive `somme_max` .
3. Quel appel de fonction doit-on faire pour résoudre le problème initial ?

# Exercice 3

*Cet exercice porte sur les arbres binaires et les arbres binaires de recherche.*

Dans cet exercice, on utilisera la convention suivante : la hauteur d'un arbre binaire ne comportant qu'un nœud est 1.

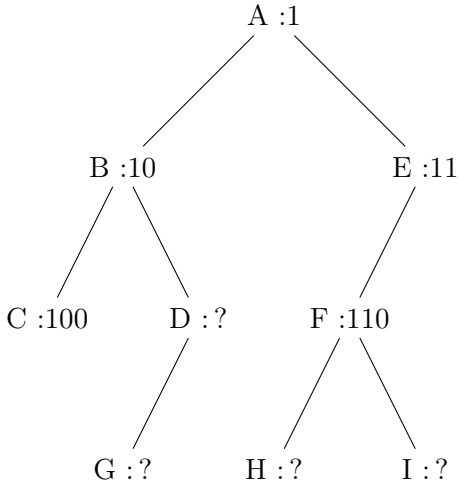
**Question 1** Déterminer la taille et la hauteur de l'arbre binaire suivant :




**Question 2** On décide de numéroter en binaire les nœuds d'un arbre binaire de la façon suivante :

- la racine correspond à 1 ;
- la numérotation pour un fils gauche s'obtient en ajoutant le chiffre 0 à droite au numéro de son père ;
- la numérotation pour un fils droit s'obtient en ajoutant le chiffre 1 à droite au numéro de son père ;

Par exemple, dans l'arbre ci-dessous, on a utilisé ce procédé pour numéroter les nœuds $A, B, C, E$ et $F$.

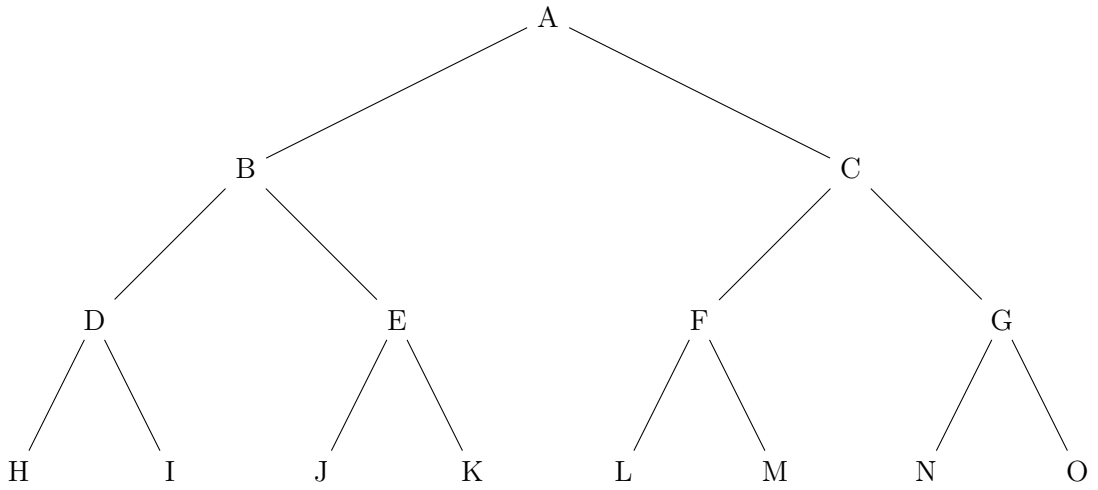



1. Dans l'exemple précédent, quel est le numéro en binaire associé au nœud $G$ ?
2. Quel est le nœud dont le numéro en binaire vaut 13 en décimal ?
3. En notant $h$ la hauteur de l'arbre, sur combien de bits seront numérotés les nœuds les plus en bas ?

4. Justifier que pour tout arbre de hauteur $h$ et de taille $n \geq 2$, on a :

$$h \leqslant n \leqslant 2^h - 1$$

**Question 3** Un arbre binaire est dit complet si tous les niveaux de l'arbre sont remplis.




*Arbre binaire complet*

On décide de représenter un arbre binaire complet par un tableau de taille $n + 1$, où $n$ est la taille de l'arbre, de la façon suivante :

— La racine a pour indice 1 ;

— Le fils gauche du nœud d'indice $i$ a pour indice $2 \times i$ ;

— Le fils droit du nœud d'indice $i$ a pour indice $2 \times i + 1$ ;

— On place la taille $n$ de l'arbre dans la case d'indice 0.

1. Déterminer le tableau qui représente l'arbre binaire complet de l'exemple précédent.
2. On considère le père du nœud d'indice $i$ avec $i \geq 2$. Quel est son indice dans le tableau ?

**Question 4** On se place dans le cas particulier d'un arbre binaire de recherche complet où les nœuds contiennent des entiers et pour lequel la valeur de chaque noeud est supérieure à celles des noeuds de son fils gauche, et inférieure à celles des noeuds de son fils droit.

Écrire une fonction `recherche` ayant pour paramètres un arbre `arbre` et un élément `element`. Cette fonction renvoie `True` si `element` est dans l'arbre et `False` sinon. L'arbre sera représenté par un tableau comme dans la question précédente.

# Exercice 4

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

L'énoncé de cet exercice utilise les mots du langage SQL suivant :

`SELECT`, `FROM`, `WHERE`, `JOIN`, `INSERT INTO`, `VALUES`, `COUNT`, `ORDER BY`.

Dans un lycée imaginaire, les données relatives aux élèves de secondes sont regroupées dans un fichier nommé `seconde_lyc.csv`. Un extrait de son contenu est représenté figure 1.

| num_eleve | nom | prenom | datenaissance | langue1 | langue2 | option | classe |
|---|---|---|---|---|---|---|---|
| 133310FE | ACHIR | Mussa | 01/01/2005 | anglais | espagnol | | 2A |
| 156929JJ | ALTMEYER | Yohan | 05/05/2005 | allemand | anglais | théâtre | 2D |
| 500633KH | BELEY | Thibaut | 05/05/2005 | anglais | espagnol | | 2A |
| 911887GA | BELEY | Marie | 05/05/2005 | anglais | espagnol | | 2A |
| 906089JJ | BELEY | Manon | 10/01/2005 | anglais | allemand | | 2E |
| 488697GA | CAILLE | Marie | 30/03/2004 | italien | anglais | | 2D |
| 193514FB | CHARPENTIER | Jules | 26/12/2005 | espagnol | anglais | | 2C |
| 321188FA | CLAUDEL | Benjamin | 09/09/2005 | espagnol | anglais | | 2E |
| 081282GF | EISEN | Carla | 23/06/2004 | anglais | allemand | | 2A |
| 026946KB | EL AYAR | Amir | 11/09/2005 | anglais | arabe | cinéma | 2D |
| 108303KG | GEHIN | Arthur | 26/02/2005 | allemand | anglais | | 2D |
| 057934BK | GROSJEAN | Alexandre | 09/11/2005 | anglais | espagnol | | 2C |
| 571113KE | HENRY | Paul | 12/03/2005 | allemand | anglais | | 2E |
| 488820DE | JACQUEY | Marc | 13/11/2005 | anglais | italien | | 2D |
| 024810CE | JULIANO | Alberto | 21/04/2005 | anglais | espagnol | | 2C |
| 249992EJ | KLEIBER | Gusti | 20/02/2005 | anglais | espagnol | cinéma | 2E |
| 492698AF | LACOUR | Julie | 06/04/2005 | italien | anglais | | 2D |
| 026454FA | LARBI | Nourdine | 14/07/2005 | espagnol | anglais | | 2C |
| 309341GD | LEFZA | Yasmina | 26/11/2005 | espagnol | anglais | | 2E |
| 076725HD | MARTIN | Victor | 13/03/2005 | anglais | espagnol | | 2A |
| 815183CB | NGUYEN | Ngong | 16/03/2005 | anglais | espagnol | | 2D |
| 094002FC | PELTIER | Romane | 14/06/2005 | allemand | anglais | | 2D |
| 321262HD | RENAULT | Zoé | 06/08/2005 | anglais | espagnol | latin | 2E |
| 075421AK | ROTH | Ursule | 03/01/2005 | anglais | allemand | | 2A |
| 121001CK | SERHANI | Sabrina | 01/09/2005 | italien | anglais | | 2D |
| 538965DJ | TIJDJANE | Yourk | 31/01/2005 | espagnol | anglais | | 2D |
| 389873GC | VIALET | Priscille | 28/02/2005 | espagnol | anglais | | 2C |
| 980306CA | WADE | Marcelin | 03/05/2005 | allemand | anglais | | 2E |
| 807158DH | WENGER | Alexandre | 20/08/2005 | allemand | anglais | | 2A |
| 666702FA | YAMAN | Elamine | 23/04/2005 | anglais | arabe | | 2D |

*Extrait du fichier* `seconde_lyc.csv`

Pour les besoins de l'organisation du lycée, le chef d'établissement exploite la base de données par des requêtes en langage `SQL`. Il a pour cela créé une table (ou relation) SQL dénommée `seconde` dans son système de gestion de bases de données dont la structure est la suivante :

| `seconde` |
|---|
| `num_eleve` (clef primaire) |
| `langue1` |
| `langue2` |
| `option` |
| `classe` |

L'attribut `num_eleve` est un entier, les autres sont des chaines de caractère (le type `CHAR`).

**Question 1**

1. Dans le modèle relationnel, quel est l'intérêt de l'attribut `num_eleve`.
2. Écrire une requête SQL d'insertion permettant d'enregistrer l'élève `ACHIR Mussa` dans la table `seconde`. Les informations relatives à cet élève sont données dans la ligne 1 du fichier `seconde_lyc.csv`.

3. Lors de l'insertion de l'élève `ALTMEYER Yohan` (ligne 2 du fichier `seconde_lyc.csv`), une erreur de saisie a été commise sur la première langue, qui devrait être `allemand`. Écrire une requête SQL de mise à jour corrigeant les données de cet élève.

**Question 2** On suppose maintenant que la table `seconde` contient les informations issues de la figure 1 (ni plus, ni moins, même si la figure 1 n'est qu'un extrait du fichier `seconde_lyc.csv`).

1. Quel est le résultat de la requête

   ```
   SELECT num_eleve FROM seconde ;
   ```
   ?
2. On rappelle qu'en SQL, la fonction d'agrégation `COUNT()` permet de compter le nombre d'enregistrements dans une table.

   Quel est le résultat de la requête

   ```
   SELECT COUNT(num_eleve) FROM seconde ;
   ```
   ?
3. Écrire la requête permettant de connaître le nombre d'élèves qui font allemand en `langue1` ou `langue2`.

**Question 3** Le chef d'établissement souhaite faire évoluer la structure de sa base de données. Pour ce faire, il créé une nouvelle table `eleve` dont la structure est la suivante :

| eleve |
|---|
| `num_eleve` (clef primaire, clef étrangère de la table `seconde`) |
| `nom` |
| `prenom` |
| `datenaissance` |

Là encore, l'attribut `num_eleve` est un entier, les autres sont des chaines de caractère (le type `CHAR`).

1. Expliquer ce qu'apporte l'information `clef étrangère` pour l'attribur `num_eleve` de cette table en termes d'intégrité et de cohérence.
2. On suppose la table `eleve` correctement créée et complétée. Le chef d'établissement aimerait lister les élèves (nom, prénom, date de naissance) de la classe 2A.

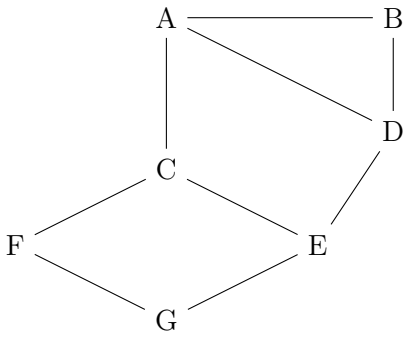
   Écrire la commande qui permet d'établir cette liste à l'aide d'une jointure entre `eleve` et `seconde`.

**Question 4** Proposer la structure d'une table `coordonnees` dans laquelle on pourra indiquer, pour chaque élève, son adresse, son code postal, sa ville, son adresse mail. Préciser la clef primaire et/ou la clé étrangère en vue de la mise en relation avec les autres tables.

# Exercice 5

*Cet exercice porte sur les réseaux en général et les protocoles RIP et OSPF en particulier.*

On considère un réseau composé de plusieurs routeurs reliés de la façon suivante :




## Le protocole RIP

Le protocole RIP permet de construire les tables de routage des différents routeurs, en indiquant pour chaque routeur la distance, en nombre de sauts, qui le sépare d'un autre routeur. Pour le réseau ci-dessus, on dispose des tables de routage suivantes :

| Table de routage du routeur A | | |
|---|---|---|
| Destination | Routeur suivant | Distance |
| B | B | 1 |
| C | C | 1 |
| D | D | 1 |
| E | C | 2 |
| F | C | 2 |
| G | C | 3 |

| Table de routage du routeur B | | |
|---|---|---|
| Destination | Routeur suivant | Distance |
| A | A | 1 |
| C | A | 2 |
| D | D | 1 |
| E | D | 2 |
| F | A | 3 |
| G | D | 3 |

| Table de routage du routeur C | | |
|---|---|---|
| Destination | Routeur suivant | Distance |
| A | A | 1 |
| B | A | 2 |
| D | E | 2 |
| E | E | 1 |
| F | F | 1 |
| G | F | 2 |

| Table de routage du routeur D | | |
|---|---|---|
| Destination | Routeur suivant | Distance |
| A | A | 1 |
| B | B | 1 |
| C | E | 2 |
| E | E | 1 |
| F | A | 3 |
| G | E | 2 |

| Table de routage du routeur E | | |
|---|---|---|
| Destination | Routeur suivant | Distance |
| A | C | 2 |
| B | D | 2 |
| C | C | 1 |
| D | D | 1 |
| F | G | 2 |
| G | G | 1 |

| Table de routage du routeur F | | |
|---|---|---|
| Destination | Routeur suivant | Distance |
| A | C | 2 |
| B | C | 3 |
| C | C | 1 |
| D | C | 3 |
| E | G | 2 |
| G | G | 1 |

**Question 1**

1. Le routeur A doit transmettre un message au routeur G, en effectuant un nombre minimal de sauts. Déterminer le trajet parcouru.
2. Déterminer une table de routage possible pour le routeur G obtenu à l'aide du protocole RIP.

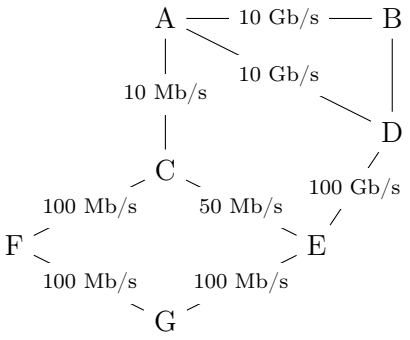
**Question 2** Le routeur C tombe en panne. Reconstruire la table de routage du routeur A en suivant le protocole RIP.

## Le protocole OSPF

Contrairement au protocole RIP, l'objectif n'est plus de minimiser le nombre de routeurs traversés par un paquet. La notion de distance utilisée dans le protocole OSPF est uniquement liée aux coûts des liaisons. L'objectif est alors de minimiser la somme des coûts des liaisons traversées.
Le coût d'une liaison est donné par la formule suivante :

$$\text{coût} = \frac{10^8}{d}$$

où $d$ est la bande passante en bits/s entre les deux routeurs.
On a rajouté sur le graphe représentant le réseau précédent les différents débits des liaisons.
On rappelle que 1 Gb/s = 1 000 Mb/s = $10^9$ bits/s.




**Question 3**

1. Vérifier que le coût de la liaison entre les routeurs A et B est 0,01.
2. La liaison entre le routeur B et D a un coût de 5. Quel est le débit de cette liaison ?

**Question 4** Le routeur A doit transmettre un message au routeur G, en empruntant le chemin dont la somme des coûts sera la plus petite possible. Déterminer le chemin parcouru. On indiquera le raisonnement utilisé.