# Mise en situation

Vous êtes mandatés pour rédiger une **étude de faisabilité** afin de développer le réseau interne de **Gearbox Québec**, une entreprise spécialisée dans le développement de jeux vidéo AAA.

En 2026, le studio aménage de **nouveaux bureaux pour l’équipe de production** au **4e étage des locaux de Gearbox Québec**.

L’espace sera complètement rénové : les murs seront retirés afin de créer un grand **espace collaboratif ouvert**, favorisant le travail en équipe des développeurs, designers et artistes.

Une image contenant logo, Graphique, capture d’écran, texte

Le contenu généré par l’IA peut être incorrect.

Source : <https://quebec.gearboxsoftware.com/>

# Contraintes du projet

**Zone principale – Aire ouverte**

* 4 îlots de 4 **stations de développement portables** (laptops de haute performance) connectées en **Wi-Fi**.
* 1 îlot supplémentaire de 2 laptops.
* Total : **18 postes sans fil** pour les développeurs.

**Salle des serveurs internes**

* **Serveur 1** : Imprimante réseau (adresse IP fixe).
* **Serveur 2** : Services **DNS et DHCP** (adresse IP fixe).

**Bureaux de gestion de projet et RH**

* 6 postes de travail fixes (PC de bureau, **connexion filaire**).

**Infrastructure réseau exigée**

* Un **routeur sans fil** pour desservir l’aire ouverte des 18 postes (équipe de production).
* Un **routeur filaire** pour desservir les 6 postes fixes (gestion & RH).
* Un **routeur frontière** pour :
  + connecter les serveurs internes (DNS, DHCP, imprimante)
  + assurer la connexion sécurisée à Internet.

**Schéma d’adressage imposé**

* **192.168.10.0 /?** → Réseau Wi-Fi (18 laptops de production).
* **192.168.20.0 /?** → Réseau filaire (6 postes fixes gestion/RH).
* **192.168.30.0 /?** → Réseau des serveurs internes.
* **199.202.106.1 /24** → Interface WAN du routeur frontière vers Internet (fournisseur = Gearbox US).

**Autres contraintes**

* Utiliser le **moins d’adresses IP possible** (optimisation des masques).
* Le schéma général du réseau doit être produit avec **Draw.io**.
* La faisabilité doit ensuite être testée avec **Cisco Packet Tracer**.
* Chaque **appareil** doit être **visible** sur les **schémas**

# Scripts Python

## Script 1 : Conversion d’adresse IP en binaire

Dans le cadre de cette partie, vous allez manipuler les adresses IPv4 afin de mieux comprendre leur structure interne.

Vous devez écrire un programme Python nommé **ipToBinary.py**. Ce programme reçoit comme **seul paramètre** une adresse IPv4 en « dot decimal notation » (ex. : 192.168.0.1).

Le programme calcule la représentation de cette adresse en binaire et affiche le résultat en conservant la séparation en quatre blocs de 8 bits par des points (« . »).

**Exemple attendu** :

192.168.0.1 -> 11000000.10101000.00000000.00000001

## Script 2 : Nombre d’adresses utilisables dans un sous-réseau

Pour cette activité, vous allez explorer la relation entre les masques de sous-réseaux et le nombre d’adresses disponibles.

Vous devez écrire un programme Python nommé **nbIPSubnet.py**. Ce programme reçoit comme **seul paramètre** un masque de sous-réseau sous forme abrégée (par ex. : /24).

Le programme calcule et affiche le **nombre d’adresses IP utilisables** dans ce sous-réseau.

**Exemple attendu** :

/24 -> 254

Modalités

* Compte pour **15% du cours**
* Peut être fait **individuellement ou en équipe de deux**
* Doit être remis zippé sur LÉA dans **Travail pratique 01. La remise comprend :**
  + **Le fichier .png nommé trpr01\_nomcomplet1\_ nomcomplet2.png (Draw.io)**
  + **Le fichier .pkt nommé trpr01\_nomcomplet1\_ nomcomplet2.pkt (Packet Tracer)**
  + **Les 2 fichiers de script Python (ipToBinary.py + nbIPSubnet.py)**
  + **La grille d’évaluation (Excel) remplie avec vos deux noms en commentaire**
* Date de remise : **voir sur Léa pour la date selon les groupes**
  + Le travail doit être remis de façon professionnelle sous peine de pénalités
  + Une rencontre peut être convoquée avec les étudiants pour vérifier la maîtrise de leur solution

# Critères d’évaluation

## Schéma Draw.io (10 pts)

1. Les secteurs/locaux sont correctement identifiés (2 pts)
2. Les postes de travail et les serveurs sont bien identifiés (noms) et séparés selon les contraintes (2)
3. Les connexions sont correctement faites entre les différents appareils (fil et sans-fil) (2)
4. Les routeurs et commutateurs doivent être correctement positionnés (2)
5. Votre schéma doit être clair, aéré et lisible (2)

## Schéma Packet Tracer (20 pts)

1. Les secteurs/locaux sont correctement identifiés (2)
2. Les postes de travail et les serveurs sont bien séparés dans les secteurs/locaux (2)­
3. Les postes de travail sont bien identifiés (noms et IP) et séparés selon les contraintes (2)
4. Les serveurs sont bien identifiés et ont des adresses IP fixes (noms et IP) (2)
5. Les connexions sont correctement faites entre les différents appareils (fil et sans-fil) (2)
6. Les routeurs doivent être correctement positionnés (2)
7. Les commutateurs doivent être correctement positionnés (2)
8. Vos configurations sont faites pour utiliser le moins d'adresses IP possible (2)
9. Les IPS et masques fournis sont tous utilisés (2)
10. Votre schéma doit être clair, aéré et lisible (2)

## Scripts Python (5 pts X2)

1. Le script lit correctement l’argument passé en ligne de commande (1)
2. Le script applique la logique de traitement attendue et obtient le bon résultat (conversion en binaire OU calcul du nombre d’IP utilisables). (2)
3. Le script affiche le résultat exactement dans le format demandé (1)
4. Le code est indenté correctement et utilise des noms de variables explicites. (1)