

## Exercice 1 : Traitement des données météorologiques

Écrire un programme en Python qui traite des données météorologiques fournies sous forme d'une chaîne de caractères, extrait la température, l'humidité, et la pression atmosphérique, puis affiche ces informations de manière claire.

### Spécifications :

1. Extraction de la température : Écrire un code qui extrait la température à partir d'une chaîne de caractères représentant une trame de données météo.
2. Extraction de l'humidité : Ajouter du code pour extraire l'humidité à partir de cette même trame.
3. Extraction de la pression atmosphérique : Ajouter du code pour extraire la pression atmosphérique à partir de la trame de données météo.
4. Affichage des résultats : Écrire le code pour afficher les valeurs de la température, de l'humidité, et de la pression sous une forme lisible.

Exemple d'une trame de données météorologiques :

```
trame_meteo = "$METEO,22.5C,55%,1013hPa,Clair"
```

### Contraintes :

- Le programme doit découper correctement la trame de données pour extraire les informations nécessaires.
- Il doit s'assurer que les données extraites sont sous le bon format (par exemple, un nombre décimal pour la température).
- Le programme doit afficher un message d'erreur si la trame ne respecte pas le format prévu.

<b>EVALUATION SYNTHESE 1</b>
<b>La programmation en Python</b>
Les boucles, la gestion des erreurs, les entrées utilisateur et les conditions

<b>BTS CIEL</b>
<b>2<sup>ème</sup> année</b>
Page 2 sur 3

## Exercice 2 : Analyseur de Paquets Réseau (Simulation)

Développer un programme interactif en Python qui simule l'analyse de paquets réseau. Le programme doit permettre à l'utilisateur de saisir des informations sur un paquet IP, puis d'afficher les détails analysés comme la source, la destination, le protocole utilisé, et l'état de sécurité du paquet.

### Fonctionnalités attendues :

#### 1. Saisie des informations du paquet :

- Le programme doit demander à l'utilisateur de saisir une adresse IP source, une adresse IP destination, et le type de protocole (TCP, UDP, ICMP, etc.).

#### 2. Vérification de la validité des adresses IP :

- Le programme doit s'assurer que les adresses IP saisies sont valides (IPv4).

#### 3. Analyse du paquet :

- Le programme doit vérifier si les adresses IP sont internes ou externes (en supposant que l'adresse interne est de la forme 192.168.x.x).
- Il doit afficher si le paquet est considéré comme sûr (si les adresses source et destination sont internes) ou potentiellement dangereux (si l'une des deux est externe).

#### 4. Affichage des résultats :

- Le programme doit afficher les informations du paquet : adresse source, adresse destination, type de protocole, et statut de sécurité (sûr ou dangereux).

### Contraintes :

- Les adresses IP doivent être vérifiées pour leur validité (IPv4).
- Le programme doit permettre à l'utilisateur d'analyser plusieurs paquets jusqu'à ce qu'il décide d'arrêter.

<b>EVALUATION SYNTHESE 1</b>
<b>La programmation en Python</b>
Les boucles, la gestion des erreurs, les entrées utilisateur et les conditions

<b>BTS CIEL</b>
<b>2<sup>ème</sup> année</b>
Page <b>3</b> sur 3

## Résultats attendus :

```

Analyse d'un nouveau paquet réseau :
Saisir l'adresse IP source : 192.168.1K.10
Adresse IP source invalide, veuillez réessayer.
Saisir l'adresse IP source : 192.168.10.10
Saisir l'adresse IP destination : 192.168.5.110
Saisir le protocole (TCP, UDP, ICMP, etc.) : ICMP

Analyse du paquet :
Adresse source : 192.168.10.10
Adresse destination : 192.168.5.110
Protocole : ICMP
Statut de sécurité : Sûr (les deux adresses sont internes)

Voulez-vous analyser un autre paquet ? (oui/non) : oui
Analyse d'un nouveau paquet réseau :
Saisir l'adresse IP source : 192.168.5.19
Saisir l'adresse IP destination : 8.8.8.8
Saisir le protocole (TCP, UDP, ICMP, etc.) : TCP

Analyse du paquet :
Adresse source : 192.168.5.19
Adresse destination : 8.8.8.8
Protocole : TCP
Statut de sécurité : Potentiellement dangereux (adresse externe détectée)

Voulez-vous analyser un autre paquet ? (oui/non) : non

```