

Mini-projet de travaux pratiques VélosDispos

Le projet est à effectuer en binôme (ou, *exceptionnellement*, en monôme) en utilisant une combinaison judicieuse de Bash et de Python. L'ensemble du code *commenté* est à déposer sur Madoc le **2 décembre 2021 à 23h55 au plus tard**, sous la forme d'une archive compressée contenant les fichiers de code source. On re-précisera en commentaires dans chaque fichier du programme les noms et prénoms de chacun-e des étudiant-e-s.

On souhaite développer l'application *VélosDispos* pour connaître en temps réel le nombre de *BiclooPlus* disponibles à une station. On va pour cela utiliser les données ouvertes fournies par [Nantes Métropole](#).

Pour réaliser ce projet, on va écrire trois programmes :

amatch.sh. Un programme Bash pour rechercher un matching approximatif d'un nom de station *BiclooPlus* dans un fichier CSV (*comma separated values*);

velosdispos.py. Un programme Python 3 pour extraire les informations de disponibilités d'un fichier au format **JSON**;

velosdispos.sh. Un programme Bash contenant l'interface avec l'utilisateur et pilotant les deux programmes précédents.

amatch.sh.

On veut autoriser une recherche approximative sur le nom de la station. On doit pouvoir écrire :

```
X1BI030@pc1> ./velosdispos.sh micheitle
```

et obtenir le nombre de vélos disponibles à la station Michelet :

```
X1BI030@pc1> ./velosdispos.sh micheitle
À la station 095-MICHELET, il reste 0 vélo(s) (mise à jour: 14 Oct 2021, 19:34:30).
```

Le programme **agrep**, disponible dans le répertoire `~goulard-f/local/bin`, fonctionne comme **grep** mais autorise aussi le matching approximatif en choisissant un score maximal d'erreur. On va écrire un programme Bash **amatch.sh** prenant en entrée un nom de station *BiclooPlus*, éventuellement approximatif, et retournant le nom complet ainsi que l'identifiant numérique de la station matchant au mieux la requête :

```
X1BI030@pc1> ./amatch.sh Mich
Station biclooPlus Michelet (n°95)
```

Pour cela, on va s'appuyer sur le fichier au format CSV décrivant toutes les stations disponible à l'adresse https://data.nantesmetropole.fr/explore/dataset/244400404_stations-velos-libre-service-nantes-metropole/download/?format=csv&timezone=Europe/Berlin&lang=fr&use_labels_for_header=true&csv_separator=%3B. Lors de son lancement, le programme **amatch.sh** devra vérifier l'existence de ce fichier dans le répertoire courant. S'il n'est pas présent, il devra le télécharger. On utilisera ensuite **agrep** pour retourner la station dont le nom matche au mieux le paramètre passé sur la ligne de commande. Le score maximal d'erreur autorisé pour **agrep** sera de 2 par défaut mais devra pouvoir être fixé à une autre valeur avec la variable d'environnement **AMATCH_DISTANCE**.

Le programme **amatch.sh** devra afficher un message d'erreur sur la sortie standard d'erreur et retourner un code d'erreur dans les cas suivants :

Code d'erreur	Erreur
1	Appel avec le mauvais nombre de paramètres
2	Impossibilité d'exécuter le programme agrep
3	Base CSV absente et impossibilité de la récupérer sur le site de Nantes Métropole

velosdispos.py

Le programme `velosdispos.py` doit prendre deux paramètres sur la ligne de commande :

- Le nom du fichier JSON contenant la base de données des disponibilités dans les différentes stations *BiclooPlus* la plus récente ;
- L’identifiant numérique de la station *BiclooPlus* demandée par l’utilisateur.

En retour, le programme doit afficher sur la sortie standard un message indiquant le nombre de vélos disponibles à la station demandée ainsi que la date de dernière remontée de l’information :

```
X1BI030@pc1> ./velosdispos.py velosdispos.json 95
À la station 095-MICHELET, il reste 19 vélo(s) (mise à jour: 15 Oct 2021, 10:47:19).
```

La date de dernière remontée est indiquée au format UTC ([Universal Time Coordinated](#)) dans le fichier JSON. Pour la transcrire dans le temps local avant affichage, on utilisera la fonction `utc2localtime()` disponible sur [madoc](#).

Un message doit être affiché sur la sortie standard d’erreur et le code d’erreur 1 doit être retourné si le programme est appelé avec un mauvais nombre de paramètres ou un numéro de station qui n’est pas un entier.

Si l’identifiant passé en paramètre ne correspond à aucune station, il faut afficher sur la sortie d’erreur standard un message en prévenant l’utilisateur et retourner le code d’erreur 2.

velosdispos.sh

Le programme `velosdispos.sh` doit prendre en paramètre un nom de station, même approximatif. Il doit ensuite utiliser `amatch.sh` pour récupérer l’identifiant numérique de la station dont le nom est le meilleur match, puis appeler `velosdispos.py` pour afficher le nombre de vélos disponibles à cette station.

Le fichier JSON de disponibilités le plus récent peut être obtenu à l’adresse https://data.nantesmetropole.fr/api/records/1.0/search/?dataset=244400404_stations-velos-libre-service-nantes-metropole-disponibilites&rows=1000. Il est mis à jour en temps réel. Pour éviter de surcharger le serveur de Nantes Métropole, le programme devra télécharger le fichier seulement s’il n’est pas déjà présent dans le répertoire courant ou si la version présente a été récupérée plus de 15 minutes auparavant. Cet temps de 15 minutes doit pouvoir être modifié en définissant la variable d’environnement `OBSOLESCENCE_TIME` avant d’appeler `velosdispos.sh`.

Le programme `velosdispos.sh` doit gérer les cas d’erreurs suivants :

Code d’erreur	Erreur
1	Appel avec le mauvais nombre de paramètres
2	Le programme <code>amatch.sh</code> n’a pas été trouvé
2	Le programme <code>velosdispos.py</code> n’a pas été trouvé
3	Pas de station trouvée avec <code>amatch.py</code>
4	Erreur dans le téléchargement de la base de disponibilités JSON