

**Projet 01** (Source de projet: Université de Paris 13):

Calcule d'un trajet de transports publics

L'objectif du projet est de développer un programme en ProLog capable d'aider un usager d'un réseau de transport à identifier le trajet lui permettant de se rendre d'une station à une autre, en respectant certaines conditions (horaire de départ, horaire d'arrivée, minimiser le nombre de correspondances ou la durée du voyage, etc.).

Les premières questions de l'énoncé sont très détaillées, tandis que celles vers la fin sont formulées de manière moins exhaustive, ce qui vise à stimuler votre créativité.

Vous devrez rendre par email d'ici le 26 novembre un lien github contenant:

- Un rapport en format **pdf** détaillant le travail effectué et les choix d'implémentation.
- Un fichier **.pl** contenant le code correspondant aux exercices des sections 2 et 3, en suivant rigoureusement l'énoncé.
- Un fichier **README** expliquant en détail comment exécuter votre programme.

Le travail est individuel.

Vous devrez faire une démonstration de votre programme et une présentation de vos choix le mercredi 29 Novembre sur vos propres PC durant la séance de TP. Il est impératif que la démonstration soit prête dès le début de la séance, avec le programme déjà compilé et sans erreurs.

**Evaluation:**

L'évaluation se fera selon les critères suivants :

- Qualité de la conception.
- Fonctionnalités réalisées.
- Qualité du code écrit, lisibilité (votre code doit être bien indenté).
- Démonstration orale.
- Qualité du rapport.

On privilégiera un projet simple dont les choix de conception sont bien motivés et qui fonctionne, plutôt qu'un projet trop ambitieux dont la structure n'est pas claire et qui ne fonctionne pas bien.

## 1 Représentation de notre réseaux de transport

L'ensemble des lignes sera défini par le prédicat suivant :

``ligne(Nom, Type, LArret, LTrajetAller, LTrajetRetour)``

où :

- **Nom** décrit le nom de la ligne, il peut être un numéro (1, 256) ou une lettre majuscule (A, B, etc.).
- **Type** décrit le type de moyen de transport, soit metro, tram, ou bus.
- **LArret** est une liste de paires [ [A1, T1], [A2, T2], ..., [An, Tn] ] où Ai décrit le nom d'un arrêt desservi par la ligne et Ti le temps nécessaire à parcourir la distance entre A(i-1) et Ai. On suppose un temps constant dans les deux directions (de A(i-1) vers Ai et de Ai vers A(i-1)).
- **LTrajetAller** est le triplet ayant comme premier élément l'horaire du premier départ de la ligne de A1 vers An, comme deuxième élément l'intervalle en minutes entre un départ et l'autre, et comme dernier élément l'horaire du dernier départ de la ligne de A1. Les horaires sont représentés sous forme d'une paire de nombres, le premier élément représentant les heures (de 0 jusqu'à 23), le deuxième élément les minutes (de 0 jusqu'à 59). Par exemple, le triplet [[5,15], 5, [1,30]] signifie qu'il y aura un départ de A1 chaque 5 minutes à partir de 05h15 et jusqu'à 1h30.
- **LTrajetRetour** est le triplet ayant comme premier élément l'horaire du premier départ de la ligne de An vers A1, comme deuxième élément l'intervalle en minutes entre un départ et l'autre, et comme dernier élément l'horaire du dernier départ de la ligne de An.

Par exemple, on peut imaginer de définir la ligne 11 du métro parisien comme suit :

...

```
ligne(11, metro,
[
[mairie_lilas, 0],
[porte_lilas, 3],
[telegraphe, 1],
[place_fetes, 1],
[jourdain, 1],
[pyrenees, 1],
[belleville, 2],
[goncourt, 2],
[republique, 3],
[arts_metiers, 2],
[rambuteau, 1],
[hotel_de_ville, 1],
[chatelet, 1]
], [[5,15],5,[1,30]], [[5,0],5,[2,0]]
)
...
```

## 2 Recherche des itinéraires

### Exercice 01:

Le but de cet exercice est de développer les outils nécessaires à manipuler les horaires. On vous suggère de représenter les horaires sous forme d'une paire de nombres [Heures, Minutes], le premier élément représentant les heures (de 0 jusqu'à 23), le deuxième élément les minutes (de 0 jusqu'à 59).

Écrire les prédicats :

- ``addh(X, M, R)``, qui est vrai quand R est l'horaire obtenu en ajoutant les minutes M à l'horaire X.

Par exemple:

...

`addh([13, 34], 30, [14, 4])` est vrai

`addh([10, 14], 25, [14, 4])` est faux

...

- ``affiche(H)``, qui affiche sur l'écran l'horaire dans un format lisible par l'utilisateur, comme par exemple, l'évaluation de `affiche([5,37])` affichera sur l'écran 05h37.

### Exercice 02:

Étant donné un ensemble de lignes, on étudie le problème de savoir si une ligne passe par deux arrêts, éventuellement en respectant un choix de l'horaire de départs ou d'arrivée.

Écrire les prédicats suivants :

- ``lig(Arret1, Arret2, Ligne)``, qui est vrai quand Ligne passe de l'Arret1 à l'Arret2.
- ``ligtot(Arret1, Arret2, Ligne, Horaire)``, qui est vrai quand Ligne part le plus tôt possible après Horaire parmi les lignes qui vont de l'Arret1 à l'Arret2.
- ``ligtard(Arret1, Arret2, Ligne, Horaire)``, qui est vrai quand Ligne arrive le plus tard possible avant Horaire parmi les lignes qui vont de l'Arret1 à l'Arret2.

### Exercice 03:

Maintenant on considère le problème de savoir s'il y a un itinéraire entre deux arrêts (éventuellement avec des échanges de moyen de transport). Les itinéraires seront représentés par des listes qui doivent contenir comme informations l'arrêt et l'horaire de départ et d'arrivée, et tous les arrêts et horaires des échanges. Étant donné un ensemble de lignes, écrire les prédicats suivants:

- ``itinTot(Arret1, Arret2, horaire, Parcours)``, qui est vrai quand Parcours décrit un itinéraire de Arret1 à Arret2 qui part le plus tôt possible après Horaire.
- ``itinTard(Arret1, Arret2, horaire, Parcours)``, qui est vrai quand Parcours décrit un itinéraire de Arret1 à Arret2 qui arrive le plus tard possible avant Horaire.

### 3 Interface avec l'utilisateur

#### Exercice 4:

Re-écrire les prédicats des exercices précédents en ajoutant des options comme le choix du réseau (ferroviaire ou bus), la préférence par rapport à la longueur du trajet, ou le nombre de correspondances.

#### Exercice 5:

Écrire une interface utilisateur dont le fonctionnement est le suivant :

- Le programme affiche les stations desservies par les transports publics.
- L'utilisateur choisit une station de départ et une station d'arrivée et les options éventuelles (choix du réseau, préférence par rapport à la longueur du trajet, au nombre de correspondances, etc.).
- Le programme affiche le ou les parcours possibles.