

- une valeur booléenne qui est égale à `True` si une solution au problème a été trouvée, à `False` sinon ;
- la liste des positions successives prises par le cavalier pour atteindre la position finale. La liste vide si aucune solution n'a été trouvée.

★ **Exercice 2.** Les voyages forment la jeunesse (6 points)

Les systèmes de divertissement à bord des avions offrent aujourd'hui des catalogues de contenus absolument gigantesques dont plusieurs centaines voire milliers de clips vidéo. Afin d'améliorer l'expérience des usagers, la compagnie Air Kifance, vous demande d'ajouter un service qui, étant donnée une durée de vol d connue, compose, pour chaque passager, une liste de lecture optimale. La liste de lecture optimale démarre exactement 8 minutes après le début du vol puis enchaîne un ensemble de clips sans interruption et s'arrête au plus tard, 15 minutes avant la fin du vol. L'arrêt se fait impérativement en fin d'un clip (pas d'interruption possible en milieu de clip, trop frustrant !). Une liste de lecture optimale va mobiliser l'attention de l'utilisateur sur la plus grande période possible, idéalement sur l'intégralité de la période d'activité du système de divertissement durant le vol.

Votre code source doit se trouver dans le fichier `P2.py`.

✓ **Question 5.** (1 pt) Écrire une fonction qui lit un fichier texte (dont le nom est donné en paramètre) contenant les informations sur l'ensemble des clips vidéos disponibles :

```
load_clips_data(clips_database_filename: str) -> Tuple[int, List[Tuple[int, str]]]
```

Son paramètre est le nom du fichier de données à lire dont le contenu est illustré à la figure 2.

```
1 40
2 190, AC/DC - You shook me all night long
3 220, MUSE - Plug in Baby
4 416, Iron Maiden - Afraid to shoot strangers
5 235, Shaka Ponk - I'm Picky
6 221, Foo Fighters - Saint Cecilia
7 232, Imagine Dragons - Walking the wire
8 161, Greenday - Boulevard of broken dreams
9 253, Rise Against - Hero of war
```

Figure 2: Dans l'exemple ci-dessus, le vol dure 40 minutes; le titre d'AC/DC dure 190 secondes, celui de MUSE, 220 secondes et ainsi de suite.

Le résultat renvoyé par la fonction est un tuple composé de :

- une valeur entière correspondant à la durée total du vol en minutes ;
- la liste des données des clips. Chaque clip est un tuple composé d'une valeur entière correspondant à la durée en secondes du clip et d'une chaîne de caractères précisant le titre du clip.

▷ **Question 6.** (5 pt) Écrire une fonction qui calcule une séquence optimale de clips pour un vol d'une durée d minutes donnée :

```
compute_optimal_playlist(flight_duration_in_minutes: int, clips: List[Tuple[int, str]])
-> Tuple[int, List[Tuple[int, str]]]
```

Ses paramètres sont:

- la durée total du vol en minutes
- la liste des données des clips. Chaque clip est un tuple composé d'une valeur entière correspondant à la durée en secondes du clip et d'une chaîne de caractères précisant le titre du clip.

Le résultat renvoyé par la fonction est un tuple composé de :

- une valeur entière indiquant à la durée totale en seconde de la liste de lecture proposée ;
- la liste des clips sélectionnés. Chaque clip est un tuple composé d'une valeur entière correspondant à la durée en secondes du clip et d'une chaîne de caractères précisant le titre du clip.

```
1 190, AC/DC - You shook me all night long
2 416, Iron Maiden - Afraid to shoot strangers
3 161, Greenday - Boulevard of broken dreams
4 253, Rise Against - Hero of war
```

Figure 3: Le total musical est de 1020 secondes (17 minutes) et correspond à la liste de lecture la plus longue possible compte-tenu des contraintes de temps de vol, des temps d'activation et de désactivation du système de divertissement à bord.