

R1.01 INITIATION AU DÉVELOPPEMENT
FEUILLE DE TD N°14 (SUITE)
Algorithmes Gloutons

Exercice 6 *Problème : planning pour un festival : implémentation en python*

Au cours d'un festival, plusieurs spectacles ont lieu, souvent sur plusieurs scènes. On veut écrire une application pour aider des festivaliers à assister à un maximum de spectacles.



Artistes	Heures																			
	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
JL Aubert																				
C Goya																				
2Be3																				
Warhole																				
Tykho Moon																				
Horus																				
KKDZO																				

```
# Exemples de modélisation de spectacles :
s1 = {'nom': 'JL Aubert', 'debut': 8, 'fin': 10}
s2 = {'nom': '2Be3', 'debut': 11, 'fin': 15}
s3 = {'nom': 'Tyko Moon', 'debut': 5, 'fin': 11}

# Exemple de modélisation de programme :
nikopol = [
    {'nom': 'JL Aubert', 'debut': 8, 'fin': 10},
    {'nom': 'JL Aubert', 'debut': 13, 'fin': 17},
    {'nom': 'JL Aubert', 'debut': 21, 'fin': 24},
    {'nom': 'C Goya', 'debut': 6, 'fin': 9},
    {'nom': 'C Goya', 'debut': 10, 'fin': 14},
    {'nom': 'C Goya', 'debut': 17, 'fin': 18},
    {'nom': '2Be3', 'debut': 11, 'fin': 15},
    {'nom': '2Be3', 'debut': 18, 'fin': 21},
    {'nom': 'Warhole', 'debut': 8, 'fin': 13},
    {'nom': 'Warhole', 'debut': 20, 'fin': 22},
    {'nom': 'Tyko Moon', 'debut': 5, 'fin': 11},
    {'nom': 'Tyko Moon', 'debut': 13, 'fin': 16},
    {'nom': 'Horus', 'debut': 7, 'fin': 18},
    {'nom': 'KKDZO', 'debut': 10, 'fin': 12}
]
```

6.1 Pourquoi ne pas modéliser le programme d'un festival par un ensemble de spectacles ?

Pour les questions suivantes, il faut compléter le fichiers `festival.py` disponible sur CELENE.

6.2 Écrire une fonction `compatibles(spectacle1, spectacle2)` qui détermine si les deux spectacles `spectacle1` et `spectacle2` sont compatibles, c'est à dire s'il est possible de les voir tous les deux.¹

6.3 Écrire une fonction `tous_compatibles(selection, spectacle)` qui détermine si le spectacle `spectacle` est compatible avec tous les spectacles de `selection` (`selection` étant une liste de spectacles).

6.4 Implémentation du premier algorithme de l'exercice précédent

- a) Écrire une fonction qui `tri_selon_debut` qui prend en paramètre un programme et qui trie les spectacles du programme selon leur heure de début croissante.
- b) Écrire une fonction `prochain_spectacle(programme, heure)` qui prend en paramètre un programme dont les spectacles sont **triés** par heure de début croissante et qui donne le premier spectacle qui commence après l'heure indiquée.
- c) Écrire une fonction `selection1(programme)` qui propose la sélection de spectacles donnée par l'algorithme 1.


6.5 Implémentation du deuxième algorithme de l'exercice précédent

- a) Écrire une fonction qui `tri_selon_duree` qui prend en paramètre un programme et qui trie les spectacles du programme selon leur durée croissante
- b) Écrire une fonction `prochain_spectacle(programme, selection)` qui prend deux paramètres : `programme` dont les spectacles sont **triés** par heure de début croissante et `selection` qui est une sélection de programmes. Cette fonction doit renvoyer le premier spectacle du programme qui est compatible avec tous les spectacles de la sélection.
- c) Écrire une fonction `selection2(programme, heure_arrivee)` qui propose la sélection de spectacles donnée par l'algorithme 2.

6.6 Implémentation du troisième algorithme de l'exercice précédent

Écrire une fonction `selection3(programme, heure_arrivee)` qui propose la sélection de spectacles donnée par l'algorithme 3.

6.7 D'après vous, lequel de ces trois algorithmes est le "meilleur" ?

1.  Indication : allez voir l'exercice 3