

Nom :	Prénom :	Classe :
-------	----------	----------

NSI 1re — DS 04

Coefficient 3	Durée 1 heure	Calculatrice interdite	Note ramenée à 20	Courage !
---------------	---------------	------------------------	-------------------	-----------

1) Cinéma - 6 pts

Un cinéma cannois propose les tarifs suivants :

- Plein tarif : 13,50 EUR
- 16 ans et moins : 6,40 EUR
- 65 ans et plus : 10,50 EUR
- Supplément salle IMAX : 6 EUR
- Supplément film en 3D : 1,50 EUR
- Supplément location de lunettes 3D : 1,50 EUR

Proposer un programme Python prenant en compte toutes ces options :

```
1  # Partie facultative
2  age = int(input('Âge svp ? '))
3  supp_imax = bool(int(input('Supplément IMAX ? (0/1) ')))
4  supp_film3D = bool(int(input('Supplément film en 3D ? (0/1) ')))
5  supp_lunettes3D = bool(int(input('Supplément location de lunettes 3D ? (0/1) ')))
6
7  # Partie requise
8  if age <= 16:
9      p = 6.4
10 elif age >= 65:
11     p = 10.5
12 else:
13     p = 13.5
14 if supp_imax == True:
15     p += 6
16 if supp_film3D == True:
17     p += 1.5
18 if supp_lunettes3D:
19     p += 1.5
20
21 print('Prix à payer :', p)
```

2) Algorithme de comptage - 5 pts

Créer le programme permettant de compter le nombre total d'éléments présents dans un tableau. *Attention : la fonction len est interdite. Décrire le fonctionnement de cet algorithme avec vos mots, puis proposer une implémentation en langage Python.*

Description du fonctionnement :

- On crée une variable compteur valant 0 et qui va servir à compter le nombre d'éléments du tableau.
- Puis, au sein d'un parcours par valeur, on incrémente la variable compteur de 1.
- À la fin, on obtient bien le nombre total d'éléments présents dans le tableau.

Implémentation Python :

```
1  compteur = 0      # Initialise le décompte à 0
2  for _ in tableau: # Lance un parcours par valeur
3      compteur = compteur + 1 # Incrémente la valeur du « compteur »
```

3) Parcours - 3 pts

Proposer deux extraits de code démontrant un parcours par valeur et un parcours par indice d'un tableau.

Parcours par valeur	Parcours par indice
<pre>for v in tableau: print(v)</pre>	<pre>for i in range(len(tableau)): print(i)</pre>

4) Algorithme accumulateur - 5 pts

Créer le programme permettant de calculer la somme (accumuler la valeur) de tous les éléments d'un tableau. Attention : la fonction `sum` est interdite. Décrire le fonctionnement de cet algorithme avec vos mots, puis proposer une implémentation en langage Python.

Description du fonctionnement :

- On crée une variable d'accumulation valant 0 et qui va servir à accumuler la valeur de chaque élément du tableau.
- Puis, au sein d'un parcours par valeur, on ajoute la valeur de chaque élément à notre variable d'accumulation.
- À la fin, on obtient bien la somme de tous les éléments du tableau.

Implémentation Python :

```
1  accumulateur = 0 # Initialise la variable à 0
2  for v in tableau: # Lance un parcours par valeur
3      # Ajoute la valeur de l'élément actuel à celle de l'accumulateur :
4      accumulateur = accumulateur + v
```

5) Algorithme de recherche d'une valeur maximale - 5 pts

Créer le programme permettant de chercher la valeur maximale présente dans un tableau. *Attention : la fonction max est interdite. Décrire le fonctionnement de cet algorithme avec vos mots, puis proposer une implémentation en langage Python.*

Description du fonctionnement :

- On crée une variable de suivi de valeur maximum qui prend comme valeur par défaut celle du premier élément du tableau.
- Puis, au sein d'un parcours par valeur, on compare la valeur de chaque élément à celle de la variable de suivi de valeur maximum. Et, si la valeur de l'élément actuel est plus élevée que celle de la variable de suivi de valeur minimum, alors on met à jour la variable de suivi de valeur maximum en lui donnant celle de l'élément actuel.
- À la fin, on obtient bien la valeur maximale présente dans le tableau.

Implémentation Python :

```
1  v_max = tableau[0] # On prend comme valeur maximale initiale
2                          # celle du premier élément du tableau
3  for v in tableau: # Lance un parcours par valeur
4      if v > v_max: # Si la valeur actuelle (celle sur laquelle on itère) est
5                  # plus élevée que la valeur maximale enregistrée jusqu'ici ...
6          v_max = v # Alors on sauvegarde cette valeur actuelle
7                  # comme nouvelle valeur maximale enregistrée
```

6) Identifier un algorithme - 3 pts

Analyser puis identifier les algorithmes ci-dessous, en les nommant convenablement :

N°	Code Python	Nom de l'algorithme
1	<pre>for i in range(len(tableau) - 1): valeur_min = tableau[i] indice_min = i for j in range(i + 1, len(tableau)): valeur_observed = tableau[j] if valeur_observed < valeur_min: indice_min = j valeur_min = tableau[j] tableau[indice_min] = tableau[i] tableau[i] = valeur_min</pre>	<p>Algorithme de tri par sélection</p> <p><i>Le tri par sélection construit le tableau trié en répétant : trouver l'élément minimal (ou maximal) dans la portion non triée et l'échanger avec la première position non triée, étendant ainsi la portion triée d'un élément.</i></p>
2	<pre>i_g = 0 i_d = len(tableau) - 1 while i_g <= i_d: i_m = (i_g + i_d) // 2 v_m = tableau[i_m] if v_cherchee < v_m: i_d = i_m - 1 elif v_cherchee > v_m: i_g = i_m + 1 else: print(i_m) break</pre>	<p>Algorithme de recherche dichotomique</p> <p><i>La recherche dichotomique trouve une valeur dans un tableau trié en comparant l'élément cherché au milieu, puis en répétant sur la moitié gauche ou droite selon le résultat jusqu'à le trouver ou réduire l'intervalle à vide.</i></p>
3	<pre>for i in range(1, len(tableau)): cle = tableau[i] j = i while tableau[j-1] > cle and j > 0: tableau[j] = tableau[j-1] j = j - 1 tableau[j] = cle</pre>	<p>Algorithme de tri par insertion</p> <p><i>Le tri par insertion parcourt un tableau de gauche à droite et, pour chaque élément, l'insère à sa place correcte dans la portion déjà triée en décalant les éléments plus grands vers la droite.</i></p>

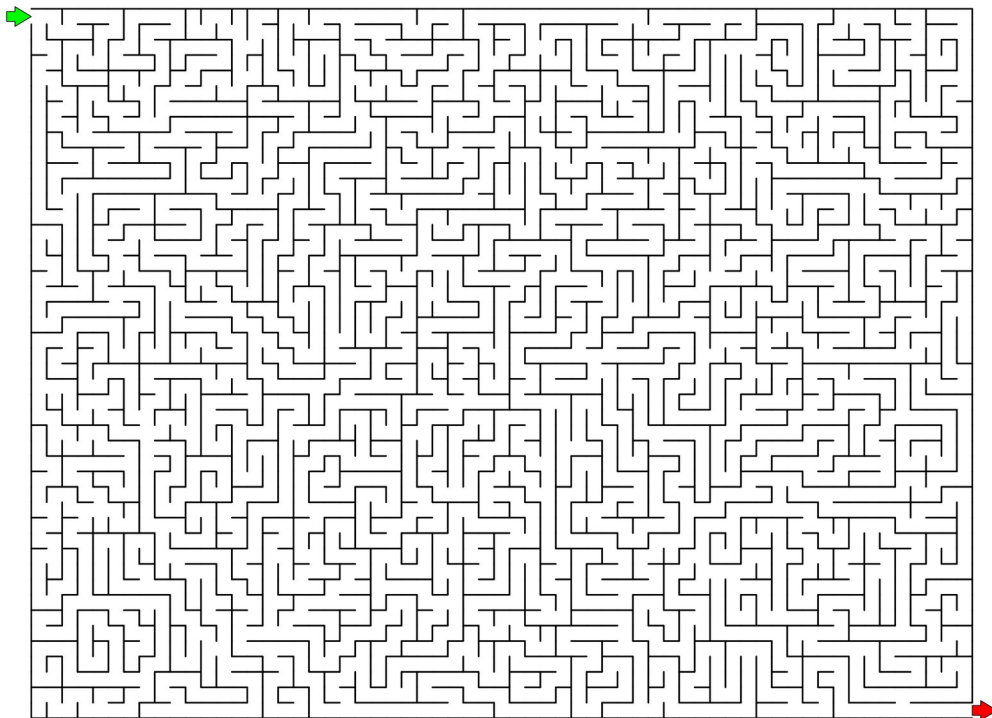
7) Algorithme de recherche de l'indice d'une valeur - 3 pts

Créer le programme permettant de rechercher l'indice de la valeur spécifique de votre choix présente dans un tableau. *On suppose la valeur spécifique comme étant unique. Attention : la fonction `index` est interdite. Proposer directement une implémentation en langage Python.*

Implémentation Python :

```
1 # On prend comme exemple la recherche de la valeur « 8 »
2
3 i_recherche = False # Par défaut, on estime que l'indice de cette valeur
4                     # n'a pas encore été trouvé dans le tableau
5 for i in range(len(tableau)): # Lance un parcours par indice
6     v = tableau[i] # On récupère la valeur d'un élément
7                 # à partir de l'indice actuel (celui sur lequel on itère)
8     if v == 8:     # Si cette valeur actuelle est celle recherchée ...
9         i_recherche = i # Alors on en sauvegarde l'indice
```

Fin



ref KDB41M9