

Recherche d'une valeur dans un tableau

1. Rechercher une valeur dans un tableau (version 1)

Énoncé	A partir d'une liste de n notes de la classe, rechercher la présence de la note 12/20.
Exemple	Liste de notes n°1 : 5,14,18,11,10,12,10,9,16 : oui la note est présente Liste de notes n°2 : 5,14,18,11,10,11,10,9,16 : non, la note n'est pas présente
Solution	Rechercher une valeur dans un tableau de dimension n. Retourner vrai si la note est présente, faux dans le cas contraire.
Entrées	Tableau de valeurs et note à rechercher.
Sortie	Valeur booléenne (vrai ou faux) représentant la présence de la valeur à chercher.
Pré-condition	Le tableau de valeurs est compris entre 0 et 20 inclus. Le tableau possède au moins 2 valeurs.
Post-condition	Un booléen vrai est retourné si la note est présente dans le tableau.

Algorithme en pseudo-code	Programme en python
Fonction rechercheNote(liste, note en décimal) valeur booléenne Pour indice de 0 longueur(Liste) -1 Si Liste[indice] = note alors Retourner vrai Fin si Fin pour Retourner faux Afficher(rechercheNote([5,14,18,11,10,12,10,9,16],12))	<pre>def rechercheNote(Liste,note): for indice in range(0,len(Liste)): return False print(rechercheNote([5,14,18,11,10,12,10,9,16],12))</pre>

Complexité de l'algorithme : O(n) car il faut parcourir le tableau de dimension n.

Résultats du test : Test effectué avec la liste : Liste=[5,18,10,12,10,14,18] et la note à rechercher 12

Étape	Note à chercher	Variable indice	Liste[indice]	Variable trouve
Avant de rentrer dans la boucle	12			Faux
Dans la boucle (1 ^{ère} itération)	12	0	5	Faux
Dans la boucle (2 ^{ème} itération)				
Dans la boucle (3 ^{ème} itération)				
Dans la boucle (4 ^{ème} itération)				
Dans la boucle (5 ^{ème} itération)				
Dans la boucle (6 ^{ème} itération)				
Dans la boucle (7 ^{ème} itération)				
En sortie de boucle				

Conclusion : L'algorithme se termine en un temps fini et produit la sortie désirée (présence ou non de la valeur recherchée).

2. Rechercher une valeur dans un tableau (version 2)

Énoncé	A partir d'une liste de n notes de la classe, rechercher la présence de la note 12/20.
Exemple	Liste de notes n°1 : 5,14,18,11,10,12,10,9,16 : oui la note est présente Liste de notes n°2 : 5,14,18,11,10,11,10,9,16 : non, la note n'est pas présente
Solution	Rechercher une valeur dans un tableau de dimension n. Retourner vrai si la note est présente, faux dans le cas contraire.
Entrées	Tableau de valeurs et note à rechercher.
Sortie	Valeur booléenne (vrai ou faux) représentant la présence de la valeur à chercher.
Pré-condition	Le tableau de valeurs est compris entre 0 et 20 inclus. Le tableau possède au moins 2 valeurs.
Post-condition	Un booléen vrai est retourné si la note est présente dans le tableau.

Algorithme en pseudo-code	Programme en python
Fonction rechercheNote(liste, note en décimal) valeur booléenne trouve ← faux indice ← 0 Tant que trouve ≠ vrai && indice < longueur(Liste) Si Liste[indice] = note alors trouve ← vrai Fin si indice ← indice+1 Fin tant que Retourner trouve Afficher(rechercheNote([5,14,18,11,10,12,10,9,16],12))	def rechercheNote(Liste,note): trouve= False indice= 0 while trouve!= True and indice<len(Liste): return trouve print (rechercheNote([5,14,18,12,10,12,10,9,16],12))

Complexité de l'algorithme : O(n) car il faut parcourir le tableau de dimension n.

Résultats du test : Test effectué avec la liste : Liste=[5,14,18,12,10,12,10,9,16] et la note à rechercher 12

Étape	Note à chercher	Variable indice	Liste[indice]	Variable trouve
Avant de rentrer dans la boucle	12	0		Faux
Dans la boucle (1 ^{ère} itération)				
Dans la boucle (2 ^{ème} itération)				
Dans la boucle (3 ^{ème} itération)				
Dans la boucle (4 ^{ème} itération)				
En sortie de boucle				

Intérêt de la boucle « while » par rapport à la boucle « for » : Le parcours du tableau pour trouver la valeur peut se terminer plus rapidement en fonction de la position de la valeur dans le tableau. Ce n'est pas le cas ici car le return True

Conclusion : On a une affectation supplémentaire et une condition supplémentaire....

3. Rechercher une valeur maximale dans un tableau

Énoncé	À partir d'une liste de n notes de la classe, rechercher la note maximale.
Exemple	Liste de notes : 5,14,18,11,10,12,10,9,16 note max : 18
Solution	Rechercher la valeur maximale dans un tableau de dimension n, puis retourner cette valeur.
Entrée	Tableau de valeurs.
Sortie	Valeur de la note maximale.
Pré-condition	Le tableau de valeurs est compris entre 0 et 20 inclus. Le tableau possède au moins 2 valeurs
Post-condition	Une valeur (entière ou décimale) maximale du tableau est retournée.

Algorithme en pseudo-code	Programme en python
Fonction maxNote(liste) valeur décimale valeur ← Liste[0] Pour indice de 1 à longueur(Liste) -1 Si Liste[indice] > valeur alors valeur ← Liste[indice] Fin si Fin pour Retourner valeur afficher(maxNote([5,14,18,12,10,12,10,9,16]))	def maxNote(Liste): print (maxNote([5,14,18,12,10,12,10,9,16]))

Complexité de l'algorithme : O(n) car il faut parcourir le tableau de dimension n.

Résultats du test : Test effectué avec la liste : Liste=[5,14,18,12,10,12,10,9,16]

Étape	Variable indice	Liste[indice]	Variable valeur
Avant de rentrer dans la boucle		5	5
Dans la boucle (1 ^{ère} itération)	1	14	14
Dans la boucle (2 ^{ème} itération)			
Dans la boucle (3 ^{ème} itération)			
Dans la boucle (4 ^{ème} itération)			
Dans la boucle (5 ^{ème} itération)			
Dans la boucle (6 ^{ème} itération)			
Dans la boucle (7 ^{ème} itération)			
Dans la boucle (8 ^{ème} itération)			
En sortie de boucle			

Conclusion : L'algorithme se termine en un temps fini et produit la sortie désirée (valeur maximale du tableau trouvée).

4. Rechercher une valeur minimale dans un tableau

Énoncé	À partir d'une liste de n notes de la classe, rechercher la note minimale.
Exemple	Liste de notes : 14,18,11,10,5,10,9,16 note min :
Solution	
Entrée	
Sortie	
Pré-condition	
Post-condition	

Algorithme en pseudo-code	Programme en python
Fonction minNote(liste) valeur décimale Retourner valeur Afficher(minNote([14,18,12,10,5,10,9,16]))	def minNote(Liste): print (minNote([14,18,12,10,5,10,9,16]))

Complexité de l'algorithme : O(n) car il faut parcourir le tableau de dimension n.

Résultats du test : Test effectué avec la liste : Liste=[14,18,12,10,5,10,9,16]

Étape	Variable indice	Liste[indice]	Variable valeur
Avant de rentrer dans la boucle		14	14
Dans la boucle (1^{ère} itération)	1	18	14
Dans la boucle (2^{ème} itération)			
Dans la boucle (3^{ème} itération)			
Dans la boucle (4^{ème} itération)			
Dans la boucle (5^{ème} itération)			
Dans la boucle (6^{ème} itération)			
Dans la boucle (7^{ème} itération)			
En sortie de boucle			

Conclusion : L'algorithme se termine en un temps fini et produit la sortie désirée (valeur minimale du tableau trouvée).

Conclusion : L'algorithme se termine en un temps fini et produit la sortie désirée (calcul de la valeur moyenne).