

Nom :

Note : / 10

Prénom :

❑ Exercice 1 : Recherche dans une liste

1. Ecrire une version *itérative* d'une fonction **recherche** qui prend en argument une liste d'entiers **l** et un entier **x** et renvoie un booléen indiquant si **x** est présent dans **l**. Par exemples :

recherche([5, 2, 5, 3, 7], 3) doit renvoyer **True**

recherche([5, 2, 5, 3, 7], 4) doit renvoyer **False**.

.....	.../2
.....	
.....	
.....	
.....	

2. Donner une version *réursive* de cette fonction.

⊗ On pourra comparer **x** avec le premier élément de **l** et appeler récursivement la recherche sur le reste de la liste si nécessaire.

.....	.../3
.....	
.....	
.....	
.....	

❑ Exercice 2

On dispose d'une liste d'entiers et on s'intéresse au problème de la recherche de la somme maximale qu'on peut former avec cette liste *sans jamais utiliser deux nombres consécutifs*. Par exemple, si on dispose de la liste [5, 2, 5, 3, 7], la somme maximale qu'on peut former est 12 (5 + 5 + 7).

1. Quelle est la somme maximale qu'on peut obtenir avec la liste [6, 7, 2, 6, 4, 3] ?

.....	.../1
.....	

2. On propose l'algorithme glouton suivant pour résoudre ce problème : si la liste n'a qu'un seul nombre alors on le prend sinon on teste quel est le plus grand des deux premiers nombres. Si c'est le premier alors on le choisit et on recommence à partir du troisième nombre, sinon on choisit le deuxième et on recommence à partir du quatrième nombre. Par exemple sur la liste [1, 3, 6, 8] on choisirait le 3 puis on recommencerait sur la liste [8] et on choisirait le 8. Prouver par un contre-exemple de votre choix que cet algorithme glouton ne donne pas forcément la solution optimale.

.....	.../1
.....	
.....	

3. Ecrire une fonction **somme_glouton** qui prend en argument une liste d'entiers et renvoie la somme obtenue en utilisant l'algorithme glouton décrit à la question précédente.

.....	.../3
.....	
.....	
.....	
.....	