

Quick – 23 février 2024 – durée 1 h

Sont interdits : les documents, les ordinateurs, les téléphones (incluant smartphones, tablettes,... tout ce qui contient un dispositif électronique).

Seuls les dictionnaires papier pour les personnes de langue étrangère sont autorisés.

Il sera tenu compte de la qualité de la rédaction et de la clarté de la présentation (2 pts).

Une phrase courte et claire vaut mieux qu'une expression longue confuse avec des ratures.

En cas d'incompréhension du sujet, préciser les hypothèses de travail que vous faites et continuer.

Le barème **indicatif** : Exercice 1 : 5 pts (5 à 10 mn); Exercice 2 : 5 pts (10 à 20 mn), Exercice 3 : 8 pts (20 à 30 mn) Relecture (~ 5mn)

Les 3 exercices sont indépendants.

I : Duplication

On considère un tableau de n éléments (type abstrait défini en amont) parmi lesquels il y a des éléments dupliqués (doublons). On souhaite identifier les éléments qui sont dupliqués. Par exemple, pour le tableau suivant, les éléments sont des caractères,

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
a	b	r	a	c	a	d	b	r	a		d	i	t	-	e	l	l	e

on construit un nouveau tableau constitué des éléments apparaissant au moins 2 fois dans le tableau donné

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
a	b	r	d	e	l	/	/	/	/	/	/	/	/	/	/	/	/	/

- I.1. Écrire un algorithme efficace qui renvoie un tableau contenant les éléments qui sont dupliqués, le tableau renvoyé ne contiendra pas de doublons, on ne pourra parcourir le tableau en entrée qu'une seule fois. Préciser avec soin les structures de données auxiliaires utilisées.
- I.2. Évaluer la complexité en moyenne de cet algorithme en nombre d'opérations effectuées (on précisera les opérations choisies pour l'évaluation).
- I.3. Évaluer la complexité en mémoire de cet algorithme.

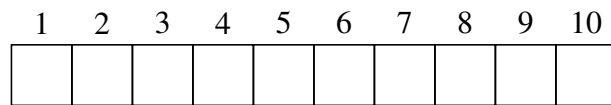
II : Une histoire de dés

On dispose d'un dé avec 6 faces et on souhaite simuler un dé à 8 faces à partir de lancers de dés à 6 faces. Pour formaliser le problème, on dispose d'une fonction $de_6()$ qui renvoie une valeur entière entre 1 et 6. On modélise une séquence d'appels à la fonction de_6 par une suite de variables aléatoires indépendantes de même loi uniforme sur $\{1, 2, 3, 4, 5, 6\}$.

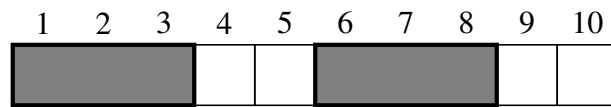
- II.1. À partir de la fonction $de_6()$, écrire une fonction $de_8()$ simulant un dé avec 8 faces équiprobables.
- II.2. Évaluer le nombre moyen d'appels à $de_6()$ pour simuler une valeur d'un dé à 8 faces. Commenter votre résultat.

III : Places de parking

On considère un parking constitué de places en ligne. On peut garer des camions sur ce parking sauf qu'ils prennent 3 places de voiture.



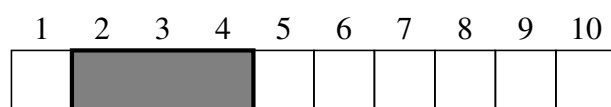
Parking vide



Parking avec 2 camions garés en position 1 et 6



Parking avec 2 camions garés en position 5 et 8



Parking avec 1 camion garé en position 2

Une configuration du parking correspond au placement de différents camions sur le parking, évidemment 2 camions ne peuvent pas occuper une même place et il n'est pas nécessaire de remplir le parking au maximum. On présente, dans les exemples ci-dessus, 4 configurations possibles pour un parking de 10 places. Ce ne sont que des exemples, il y a d'autres configurations, un parking de 10 places permettant de garer au maximum 3 camions.

On note n la longueur du parking en nombre de places et C_n le nombre de configurations possibles du parking.

- III.1. Proposer un algorithme pour calculer C_n et calculer C_{10} . Pour cela on pourra écrire une équation de récurrence vérifiée par les C_n . Pour construire cette équation de récurrence considérer la place 1 du parking elle est soit occupée par un camion soit elle n'est pas occupée par un camion.
- III.2. Évaluer la complexité de votre algorithme et commenter cet algorithme.
- III.3. Écrire un algorithme qui énumère toutes les configurations possibles d'un parking de taille n .
- III.4. Rédiger la preuve de cet algorithme.
- III.5. Que faudrait-il modifier à cet algorithme pour énumérer les configurations ayant exactement 2 camions ?