

INFO TP 9 & 10 : ENTRAÎNEMENT EXOS ORAL

Exercice 1 :

Soit n un entier vérifiant $n \leq 26$. On souhaite écrire un programme qui code un mot en décalant chaque lettre de l'alphabet de n lettres.

Par exemple, pour $n = 3$, le décalage sera le suivant :

avant décalage	a	b	c	d	...	...	x	y	z
après décalage	d	e	f	g	...	...	a	b	c

Le mot `oralensam` devient ainsi `rudohqvdp`.

1. Définir une chaîne de caractères contenant toutes les lettres dans l'ordre alphabétique (caractères en minuscule).
2. Écrire une fonction `décalage`, d'argument un entier n , renvoyant une chaîne de caractères contenant toutes les lettres de l'alphabet dans l'ordre alphabétique, décalées de n , comme indiqué ci dessus.
3. Écrire une fonction `codage` d'arguments un entier n et une chaîne de caractères `phrase` renvoyant `phrase` codée avec un décalage de n lettres.
4. Comment peut-on décoder un mot codé ?

Exercice 2 :

Soit un entier naturel n non nul et une liste `t` de longueur n dont les termes valent 0 ou 1. Le but de cet exercice est de trouver le nombre maximal de 0 contigus dans `t` (c'est-à-dire figurant dans des cases consécutives). Par exemple, le nombre maximal de zéros contigus de la liste `t1` suivante vaut 4 :

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
t1[i]	0	1	1	1	0	0	0	1	0	1	1	0	0	0	0

1. Écrire une fonction `nombreZeros(t, i)`, prenant en paramètres une liste `t`, de longueur n , et un indice i compris entre 0 et $n - 1$, et renvoyant :

$$\begin{cases} 0, & \text{si } t[i] = 1 \\ \text{le nombre de zéros consécutifs dans } t \text{ à partir de } t[i] \text{ inclus, si } t[i] = 0 \end{cases}$$

Par exemple, les appels `nombreZeros(t1, 4)`, `nombreZeros(t1, 1)` et `nombreZeros(t1, 8)` renvoient respectivement les valeurs 3, 0 et 1.

- Comment obtenir le nombre maximal de zéros contigus d'une liste `t` connaissant la liste des `nombreZeros(t,i)` pour $0 \leq i \leq n-1$?
En déduire une fonction `nombreZerosMax(t)`, de paramètre `t`, renvoyant le nombre maximal de 0 contigus d'une liste `t` non vide. On utilisera la fonction `nombreZeros`.
- Quelle est la complexité de la fonction `nombreZerosMax` construite à la question précédente ?
- Trouver un moyen simple, toujours en utilisant la fonction `nombreZeros`, d'obtenir un algorithme plus performant.

Exercice 3 : Matrices magiques :

Une matrice carrée d'ordre n est dite magique si elle contient tous les nombres de 1 à n^2 , et si les sommes des nombres de chaque ligne, de chaque colonne et de chaque diagonale sont toutes égales à une même constante s .

- Au brouillon, exprimer la constante s en fonction de n .
- Créer une fonction `EstMagique`, d'argument une matrice T qui renvoie un booléen indiquant si T est magique ou pas.

Tester cette fonction sur les matrices :

$$A = \begin{pmatrix} 4 & 9 & 2 \\ 3 & 5 & 7 \\ 8 & 1 & 6 \end{pmatrix} \quad \text{et} \quad B = \begin{pmatrix} 1 & 8 & 2 \\ 4 & 5 & 7 \\ 6 & 9 & 3 \end{pmatrix}$$

Une méthode permettant de construire une matrice magique de taille impaire $n = 2p + 1$ est la suivante :

- On construit une matrice de taille n remplie de zéros. On considère cette matrice comme la représentation sur une période d'une matrice infinie n -périodique en lignes et en colonnes.
 - On remplace ensuite les zéros de la matrice avec les nombres de 1 à n^2 comme suit :
 - on met le 1 dans la case située sous la case centrale de la matrice ;
 - on place ensuite chaque nombre de 2 à n^2 dans la case située ligne suivante et colonne suivante de celles où on a mis le nombre précédent. Si cette case est déjà remplie, on avance encore d'une ligne et on recule d'une colonne.
- On admet que la matrice ainsi construite est magique.
- Construire à la main, selon cette méthode, une matrice magique d'ordre 3.
 - Écrire une fonction `Magique` d'argument un entier p qui renvoie la matrice magique de taille $n = 2p + 1$ créée à l'aide de la méthode précédente.