

INFO-H-100 – Informatique – Partie Programmation – Prof. Th. Massart  
1<sup>ère</sup> année du grade de Bachelier en Sciences de l'Ingénieur  
Examen de première session

---

### Remarques préliminaires

- On vous demande de répondre à chaque question sur une feuille séparée.
- N'oubliez pas d'inscrire votre nom, prénom et numéro de matricule sur chaque feuille.
- Vous disposez de 2 heures et vous ne pouvez pas utiliser de notes.
- La réponse à la question doit comprendre le code *C++* structuré et conforme aux règles de bonne pratique et conventions, toutes les déclarations associées et des commentaires pertinents.
- Vous pouvez ajouter des fonctions si cela vous semble nécessaire.
- Sauf mention contraire, vous ne pouvez utiliser aucune fonction de bibliothèques.

### Question 1 - 1337 5p34k (6 points)

Soit une chaîne de caractères `phrase` contenant au maximum 128 caractères et un vecteur `alphabet` contenant 26 caractères. On désire transformer la chaîne de caractères `phrase` en remplaçant chacun de ses caractères par le caractère correspondant dans le vecteur `alphabet`. Ainsi, `alphabet[0]` contiendra le caractère remplaçant la lettre `a` ou `A`. Seules les 26 lettres de l'alphabet devront être remplacées qu'elles soient en minuscule ou en majuscule. Les autres caractères de la chaîne `phrase`, comme les espaces, resteront inchangés.

On vous demande d'écrire une fonction `leetSpeak` qui recevra les deux vecteurs `phrase` et `alphabet` en paramètre et qui réalisera la traduction décrite ci-dessus.

Exemple :

`phrase`

L	e	e	t		s	p	e	a	k	\0	...
---	---	---	---	--	---	---	---	---	---	----	-----

`alphabet`

4	8	[	)	3	F	6	#	1	J	k	1	M	N	0	p	Q	2	5	7	μ	V	W	X	j	5
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Après l'appel de la fonction `leetSpeak`, la chaîne de caractères `phrase` aura la forme suivante :

`phrase`

1	3	3	7		5	p	3	4	k	\0	...
---	---	---	---	--	---	---	---	---	---	----	-----

### Question 2 - Tri d'un dictionnaire (8 points)

On vous demande d'écrire un fonction `triDico` qui reçoit les paramètres ci-dessous.

- Un tableau de caractères à deux dimensions appelé `dico`. Chaque ligne de ce tableau contient un mot. Ces mots sont rangés selon leur taille : on trouvera d'abord les mots d'une lettre puis les mots de deux lettres et ainsi de suite. Les mots contiennent au maximum `N` lettres où `N` est une constante globale donnée et ne contiennent que des majuscules entre `A` et `Z` sans accent. Un exemple se trouve sur la page suivante.
- Un vecteur `index` contenant `N+1` entiers. Le premier mot de taille 1 de `dico` se trouvera à la ligne d'indice `index[0]`, le premier mot de taille 2 se trouvera à l'indice `index[1]` et plus généralement le premier mot de taille `i` se trouvera à l'indice `index[i-1]`. Par conséquent, le dernier mot de taille 1 se trouvera à l'indice `index[1]-1` et plus généralement le dernier mot de taille `i` se trouvera à l'indice `index[i]-1`.
- Un entier `taille` indiquant la taille des mots à trier.

Votre fonction doit trier par ordre alphabétique les mots du tableau `dico` ayant une taille égale à `taille` en utilisant *le tri par sélection*. Les mots de taille différente ne bougent pas.

Exemple pour N = 4 :

dico	0	?	?	?	index	0	2	4	8	9
	A	?	?	?						
	B	U	?	?						
	A	N	?	?						
	B	I	O	?						
	B	A	C	?						
	M	U	R	?						
	A	N	S	?						
	M	U	L	E						

Après l'appel de `triDico(dico, index, 3)`, le tableau `dico` aura la forme suivante :

dico	0	?	?	?
	A	?	?	?
	B	U	?	?
	A	N	?	?
	A	N	S	?
	B	A	C	?
	B	I	O	?
	M	U	R	?
	M	U	L	E

On peut remarquer que les mots du tableau `dico` ne se terminent pas par le caractère `\0`.

### Question 3 - Série numérique (6 points)

La série

$$\sum_{n \geq 2} \frac{1}{n^\alpha}$$

converge pour  $\alpha > 1.0$ . Pour une valeur `cible` entre 0.1 et 1.0, il existe un  $\alpha$  tel que la série ci-dessus converge vers `cible`.

Etant données les constantes globales `eps` (précision) et `nMax`, on vous demande d'écrire la fonction `double alpha(double cible)` qui calcule la valeur de  $\alpha$  qui fait converger vers la valeur `cible` donnée en paramètre (à `eps` près) la série :

$$\sum_{n=2..nMax} \frac{1}{n^\alpha}$$

On suppose que `cible` est une valeur entre 0.1 et 1.0 et que le résultat  $\alpha$  sera compris entre 1.0 et 100.0. On remarque que plus  $\alpha$  est grand, plus la série a une petite valeur. La recherche se fera par *dichotomie* jusqu'à obtenir un résultat suffisamment proche pour la série.

**Rappel.** Une *recherche dichotomique* d'une valeur  $x$  dans un intervalle trié  $[bi, bs]$  compare  $x$  avec la valeur  $m$  au milieu de l'intervalle. Si la recherche n'est pas fructueuse, on réitère la recherche avec la première partie  $[bi, m]$  de l'intervalle ou la seconde  $[m, bs]$  suivant l'endroit où  $x$  est potentiellement présent.

**Remarque.** Vous pouvez utiliser les fonctions

- `double exposant(int base, double exp)` qui renvoie la valeur  $base^{exp}$
- `double abs(double a)` qui renvoie la valeur  $|a|$ .

**Conseil.** Décomposez le problème en deux fonctions :

- la première qui calcule la série pour un  $\alpha$  donné et
- la seconde, `double alpha(double cible)`, qui effectue la recherche dichotomique en utilisant la première fonction.

BON TRAVAIL !