## Contrôle de Info C

Durée :	0h45		Aucun document autorisé
Nom:		Prénom :	
			iste d'éléments de type $\underline{T}$ par une struc- déclaration de type suivante :
T elt	; ${f ct}$ noev	ct noeud { id *suivant;	
		ue la fonction longue at, ce qui est coûteux	our devait parcourir toute la liste pour
		aration du type LISTE onnerez sa déclaration	précédente afin de mémoriser la longueur complète.
cice 1, éc paramètr	rivez la f		ISTE que vous venez de définir à l'exer- renvoie la longueur d'une liste passée en livant :
/*     * Af	firmatio	ons à insérer	
•	ngueur (	LISTE 1) {	

En TD, nous avons vu que les fonctions d'insertion et de suppression d'un élément dans une liste simplement chaînée devaient traiter le cas particulier d'une insertion ou d'une suppression au rang 1. Pour éviter cet écueil, et ne plus avoir à faire de test, la structure chaînée possèdera systèmatiquement, en tête, un nœud factice. Ainsi, une liste vide (donc longueur 0) sera représentée, par une structure chaînée qui comportera ce nœud factice. Comprenez bien que d'un point de vue fonctionnelle, un élément au rang 1 est toujours au rang 1, un élément au rang 2 est toujours au rang 2, etc. Ce n'est que la représentation interne qui change.

▶ 3. Écrivez la <u>procédure</u> qui initialise son paramètre à la liste vide. Avec <u>votre</u> type LISTE de <u>l'exercice</u> 1, cette procédure possède l'en-tête suivant :

/* *	Affirmations à insér			
*/	Affirmations a inser	· E 1		
	initListe(LISTE *	1)		

▶ 4. En gérant l'élément factice de tête de liste, écrivez la procédure insèrer qui insère dans une liste 1 (de votre type LISTE) au rang r un élément e. Votre procédure devra évidement avoir exactement le même en-tête que celui de la feuille de TD (au nom forme près), et la même sémantique. Vous donnerez l'antécédent et le conséquent de votre procédure. Constatez que son écriture est simplifiée.


 occurres initi	iste, longueur	et inserer.	