

Un tableau non trié d'entiers  $E[0], \dots, E[n-1]$  est donné. Le problème est de calculer la longueur de la séquence croissante la plus longue d'éléments. Dans cette séquence, tout élément (sauf le dernier) est inférieur ou égal à son élément suivant, non forcément consécutif. Pour ce travail, vous pouvez supposer pour simplifier que tous les éléments sont différents.

Par exemple, si  $E = [0; 300; 100; 200; 1000; 400; 500; 1100; 900; 800; 600; 700; -100]$ , alors la séquence croissante la plus longue possède 7 éléments. Il s'agit de  $[0; 100; 200; 400; 500; 600; 700]$ .

Il existe une méthode en temps polynomial ( $O(n^2)$ ) qui permet de calculer la longueur de la séquence croissante la plus longue qui se termine (et utilise ou non)  $E[i]$  récursivement à partir de celles qui se terminent en  $E[0], E[1], \dots, E[i-1]$ .

Soit  $LT[i]$ , la longueur de la séquence croissante la plus longue qui se termine (et utilise)  $E_i$ .

au départ  $LT[i] = 1$

pour  $i \in [1..n]$   $LT[i] = LT[k] + 1$  pour  $k \in [0..i-1]$  tel que  $E[k] < E[i]$  et  $LT[i] < LT[k] + 1$

Exemple :

i	0	1	2	3	4	5	6	7	8	9	10	11	12
E[i]	0	300	100	200	1000	400	500	1100	900	800	600	700	-100
LT[i]	1	2	2	3	4	4	5	6	6	6	6	7	1

**Question 1** (1 pt) : Construire un nouveau projet nommé « TPxxx » où xxx est votre nom de famille et recopier dans le dossier « src » de votre projet, le fichier « Sequence.java » qui se trouve sur le serveur à l'endroit indiqué par votre surveillant.

**Question 2** (5 pts) : Compléter la méthode `initLT(int n)` proposée dans le fichier « sequence.java » dont l'objectif est de calculer la longueur de la séquence croissante la plus longue qui se termine à chacun des éléments du tableau. La valeur passée en paramètre, à la méthode `initTL`, représente la taille du tableau.

**Question 3** (2 pts) : Compléter le programme principal pour que ce dernier exécute la méthode `initLT` et affiche sur une première ligne le contenu de `E` et sur la ligne suivante le contenu de `LT`. Si vous n'avez pas réussi à rédiger la méthode « `initTL` », vous pouvez faire appel à la méthode « `initTLsecours` » pour pouvoir continuer le travail pratique demandé.

**Question 4** (3 pts) : La méthode « `affichSeq(int l)` » affiche la première séquence croissante de longueur `l` (valeur passée en paramètre) si elle existe. Elle fournit aussi un résultat dans le cas où la valeur transmise en paramètre n'existe pas. Modifier la méthode pour que, dans ce cas, le message soit « aucune séquence de cette longueur ».

Exemple : affichSeq(6) donnera l’affichage suivant : 0 100 200 400 500 1100

i	0	1	2	3	4	5	6	7	8	9	10	11	12
E[i]	0	300	100	200	1000	400	500	1100	900	800	600	700	-100
	1		2	3		4	5	6					

qui correspond aux valeurs 1, 2, 3, 4, 5 et 6 de LT en choisissant toujours la dernière occurrence en cas de valeurs multiples.

**Question 5** (3 pts) : Améliorer la méthode récursive « affichSeqRec (int li, int lg) » afin d’optimiser la recherche des valeurs égales ou supérieures à la valeur recherchée. Vous pourrez vous appuyer sur l’exemple proposé pour réaliser cette optimisation.

**Question 6** (5 pts) : En vous inspirant de la méthode précédente, rédiger la méthode récursive « String : En vous inspirant de la méthode précédente, rédiger la méthode récursive « String affichRecTous(int li, int lg, String res) » qui affiche toutes les séquences croissantes de longueur maximale lg. Cette méthode est appelée par la méthode « String affichSeqTous(int l) » donnée ci-dessous :

```
public static String affichSeqTous(int lg)
{ assert (lg>=0 && lg<E.length);
  return affichRecTous(1, lg, "sequence de taille "+lg+"\n"); }
```

Exemple : affichSeqTous(6) donnera les affichages successifs suivants (aux espaces près) :

i	0	1	2	3	4	5	6	7	8	9	10	11	12
E[i]	0	300	100	200	1000	400	500	1100	900	800	600	700	-100
	1		2	3		4	5	6					

sequence de taille 6

0 100 200 400 500 1100

	1		2	3		4	5		6				
--	---	--	---	---	--	---	---	--	---	--	--	--	--

sequence de taille 6

0 100 200 400 500 900

	1		2	3		4	5			6			
--	---	--	---	---	--	---	---	--	--	---	--	--	--

sequence de taille 6

0 100 200 400 500 800

	1		2	3		4	5				6		
--	---	--	---	---	--	---	---	--	--	--	---	--	--

sequence de taille 6

0 100 200 400 500 600

**Question 6** (1 pt) : Constituer une archive au format zip de votre dossier « TPxxx » que vous nommerez « TPxxx » où xxx est votre nom de famille. Envoyer un message à [dominique.michelucci@u-bourgogne.fr](mailto:dominique.michelucci@u-bourgogne.fr) en joignant votre archive et mettre en copie le surveillant (son adresse électronique est notée au tableau).