Exercice 1 : *Déclarer des tableaux*

Q 1 . On veut écrire un programme pour gérer des budgets mensuels, pour cela on va déclarer un type `T_BUDGET` qui sera un type tableau de douze cases indexées par les douze mois de l'année, chaque case contenant un nombre réel représentant les dépenses (ou recettes) du mois, ainsi dans le programme on pourra déclarer des variables du style :

```
var depensesVoiture, depensesMaison, depensesLoisirs, recettesSalaires : T_BUDGET ;
```

Faites la déclaration pour ce type `T_BUDGET`. (on pourra donner deux solutions dépendant de la déclaration du type pour représenter les mois).

Q 2 . Au début de chaque année on initialise toutes les variables correspondants aux différents budgets en mettant 0 dans toutes les cases.

Écrivez une procédure `initialiserBudget` qui réalise cette opération sur son unique paramètre `b` de type `T_BUDGET`.
(*Remarque* : vous essaierez de faire une version qui soit correcte indépendamment de la version choisie pour déclarer `T_BUDGET`).

Q 3 . De même à la fin de chaque année on veut calculer le "budget mensuel moyen" pour chaque catégorie.

Écrivez une fonction `budgetMoyen` qui calcule ce budget moyen à partir de son unique paramètre `b` de type `T_BUDGET` .
(Même remarque que pour la question précédente)

On veut maintenant travailler avec des représentations d'objets en machine. On sait que ces objets sont representés sur N octets (par exemple les `INTEGER` sont codés en $PASCAL$ sur 4 octets).

Q 4 . Faites les déclarations nécessaires pour le type `OCTET` qui sera un tableau de huit `BIT`

Q 5 . Que prendre comme représentation pour un `BIT` ?

Q 6 . Donnez deux manières de représenter un `OBJET` codé par N `OCTET`.

Exercice 2 : Des manipulations standards sur les tableaux

Dans tout cet exercice on suppose que le type `TABLEAU` a été déclaré à l'aide des déclarations suivantes :

```
type INDICE =  ..... ;    // un intervalle d'entiers
     ELEMENT = ........ ; // un type auquel on peut appliquer la relation d'ordre <=
     TABLEAU = array[INDICE] of ELEMENT ;
```

Q 1 . Réalisez les fonctions suivantes :

max	:	TABLEAU	$\longrightarrow$	ELEMENT
		t	$\longmapsto$	le plus grand élément présent dans t
secondMax	:	TABLEAU	$\longrightarrow$	ELEMENT
		t	$\longmapsto$	le second plus grand élément présent dans t
miroir	:	TABLEAU	$\longrightarrow$	TABLEAU
		t	$\longmapsto$	le tableau avec les mêmes valeurs que t mais dans l'ordre inverse

Q 2 . Réalisez une procédure `retourner` à un paramètre `t` de type `TABLEAU` qui transforme le tableau `t` en son miroir. On pourra en faire deux versions, la première utilisant la fonction `miroir` de la question précédente, la seconde sans ; quelle est la méthode la plus efficace ?

Exercice 3 : *Le crible d'Érathostène*

C'est une méthode très efficace pour chercher tous les nombres premiers inférieurs à un certain N. Voici l'algorithme :

- On écrit les nombres de 2 à N (on sait que 0 et 1 ne sont pas premiers)
  2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, ..., N
- On sait que 2 est premier, donc on le garde mais tous ses autres multiples ne le sont pas donc on les barre
  2, 3, ~~4~~, 5, ~~6~~, 7, ~~8~~, 9, ~~10~~, 11, ~~12~~, 13, ~~14~~, 15, ~~16~~,..., N

- On passe au suivant, 3, il n'est pas barré donc il est premier, et on peut barrer tous ses autres multiples. Le premier à barrer serait 6 mais il est déjà barré donc le premier que l'on va barrer est 9

  2, 3, ~~4~~, 5, ~~6~~, 7, ~~8~~, ~~9~~, ~~10~~, 11, ~~12~~, 13, ~~14~~, ~~15~~, ~~16~~,..., N

- On passe au suivant, 4, il est barré donc pas premier et il n'y a rien à faire

- Et on continue ainsi ...

Quelques questions sur l'algorithme :

Q 1 . Quand on traite un nombre p qui n'est pas barré donc premier, quel est le premier multiple que l'on va devoir barrer ?

Q 2 . Quand sait-on que le travail est terminé, i.e. on ne barrera plus aucun nombre et donc tous ceux qui ne sont pas barrés sont premiers ?

Nous allons maintenant écrire le programme en *Pascal*.

Q 3 . Comment représenter l'ensemble des nombres de 2 à N avec le fait qu'ils peuvent être barrés ou non ? Faites les déclarations pour cette structure de données.

Q 4 . Au départ, aucun nombre n'est barré donc on doit écrire une procédure `initialiser`. Écrivez-la.

Q 5 . Ensuite on doit traiter les nombres non barrés en barrant tous leurs autres multiples. Écrivez une procédure dont l'entête serait :

```
procedure traiterNombre(? t : TAB_PREMIERS ; ? n : CARDINAL) ;
```

`t` étant la structure de données et `n` le nombre que l'on traite. (on complétera les deux ?)

Q 6 . Écrivez maintenant la procédure

```
procedure traiterTout(? t : TAB_PREMIERS) ;
```

`t` étant la structure de données. (on complétera le ?)

Q 7 . Enfin écrivez la procédure

```
procedure afficherPremiers(? t : TAB_PREMIERS) ;
```

qui affiche les nombres premiers en supposant que `t` a été traité avant.

En TP vous réaliserez le programme complet