

Nom : .....  Prénom : .....	Note : ..... / 10
-----------------------------------	-------------------

□ **Exercice 1** : *Nombre harshad*

Les questions sont *indépendantes*, et on pourra toujours utiliser une fonction demandée à une question précédente même si cette question n'a pas été traitée. On supposera toujours déjà incluses les bibliothèques usuelles du langage C.

On dit qu'un entier strictement positif est un *nombre harshad* (ou nombre de Niven) lorsqu'il est divisible par la somme de ses chiffres dans une base donnée. Dans cet exercice, on s'intéresse aux nombres harshad en base 10. Par exemples, 48 est un nombre harshad puisque 48 est divisible par 12, de même 63 (divisible par 9) ou encore 190 (divisible par 10) sont aussi des nombres harshad. Par contre 28, ou encore 104 ne sont pas des nombres harshad.

1. Ecrire une fonction de signature `bool est_harshad(int n)` qui renvoie `true` si et seulement si `n` est un nombre harshad.

[illegible]

2. Ecrire une fonction `main` qui prend un argument en ligne de commande une chaîne de caractère, la convertit en entier (avec la fonction `atoi`) puis affiche `true` si cet entier est harshad et `false` sinon. Si aucune argument n'est donné, on affiche un message d'erreur. Par exemple, voici des comportements attendus en supposant que l'exécutable s'appelle `harshad.exe`

```
./harshad.exe 42
true
./harsahd.exe 53
false
./harshad.exe
Utilisation : ./harshad.exe <entier positif>
```

[illegible]

3. Ecrire une fonction `int* get_harshad(int limit, int *nb)` qui prend en argument un entier `limit`, et renvoie un tableau contenant les nombres de harshad inférieurs ou égaux à `limit`. De plus `*nb` doit contenir après appel le nombre de nombres harshad inférieurs ou égaux à `limit`. Par exemple comme les nombres harshad inférieurs ou égaux à 20 sont {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 18, 20} l'appel `get_harshad(20, &n)` renvoie un tableau contenant ces entiers et après l'appel `n` contient la taille de ce tableau donc 13.

.....	.../4
.....	
.....	
.....	
.....	
.....	
.....	
.....	
.....	
.....	
.....	
.....	
.....	

4. On recherche à présent des suites de nombres consécutifs qui sont tous harshad, par exemple les nombres 110, 111 et 112 sont 3 nombres harshad consécutifs. En écrivant un programme, déterminer le plus petit entiers  $h$  tel que  $h, h + 1, h + 2, h + 3$  et  $h + 4$  soient tous des nombres de harshad.