

## *TD machine n°2*

### Récurtivité sur des entiers.

**Exercice 1.** Écrire une fonction récursive qui prend en paramètre un entier `n` et qui renvoie le produit des chiffres qui le composent.

**Exercice 2.** Écrire une fonction qui prend en paramètre un entier `n` et qui renvoie un entier résultant de l'ordonnancement des chiffres de `n`. Par exemple, si l'argument de la fonction est l'entier 3714, alors le résultat est 1347.

### Récurtivité sur des chaînes de caractères.

**Exercice 3.** Écrire une fonction `rep : int -> string -> string` telle que `rep n s` renvoie la chaîne de caractères `s` répétée `n` fois. Si `n` est négatif, la fonction renvoie la chaîne vide.

**Exercice 4.** Écrire un sous-programme `sablier` qui prend en argument une chaîne de caractères dont on suppose que sa longueur est impaire, et qui l'affiche sous la forme d'un sablier. Par exemple, pour la chaîne de caractères "bonjour", le sous-programme `sablier` affiche :

```
bonjour
  onjou
    njo
      j
    njo
  onjou
bonjour
```

**Exercice 5.** Partant de la chaîne de caractères "1", on souhaite obtenir et afficher une suite de mots en comptant les chiffres successifs du mot précédent. Le deuxième mot est "11" parce qu'il y a un 1 dans "1". Ensuite, il y a deux 1, donc le troisième mot est "21" etc ... Écrire un sous-programme récursif qui étant donné un entier `n` saisi au clavier, affiche les `n` premiers mots obtenus chacun sur une ligne. Exemple, si `n = 8`, on affiche :

```
1
11
21
1211
111221
312211
13112221
1113213211
```

### Chaîne de caractères et listes

Écrire une fonction `string_split` qui prend en argument deux chaînes de caractères `s` et `c` où `c` est formée d'un seul caractère et qui retourne une liste de chaîne de caractères contenant les morceaux de `s` "coupée" aux endroits où apparaît la chaîne `c`. Exemple :

```
1 let test = "on fait un td de quoi deja?" ;;
2 let l = string_split test " " ;;
3 val l : string list = ["on"; "fait"; "un"; "td"; "de"; "quoi"; "deja?"]
```