

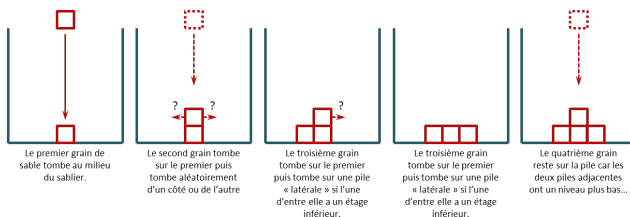
## TD – 01

## Exercices d'applications

## Exercice 1 - Le marchand de sable

On se propose de modéliser la constitution d'un tas de sable ainsi que l'écoulement des grains dans un sablier. Afin de simplifier le problème, on se restreindra à travailler en 2 dimensions. Le tas sera modélisé par une pile de grains de sable.

Dans le cas du sablier, les grains tombent toujours sur la même pile. Le processus de constitution de la pile est le suivant :



## Gestion d'une pile de sable

**Objectif** Modéliser l'écoulement d'un grain de sable sur une colonne.

Une pile de sable est modélisée par... une pile! Cette dernière est implémentée sous forme d'une liste. La taille des piles est de dimension finie notée `ht`. Une pile de 3 grains de sable dans une pile de taille 5 sera modélisée par la liste `['*', '*', '*', '', '']`.

**Question 1** Donner l'implémentation des fonctions élémentaires permettant de gérer une pile dans Python à savoir les fonctions `creer_pile`, `empiler`, `depiler`, `est_vide`. **Pour cette question on s'autorise l'utilisation des méthodes sur les listes. Vérifier le bon comportement de vos fonctions.**

**Question 2** Donner l'implémentation de la fonction `taille_pile`, permettant de connaître la taille d'une pile. **Pour cette question seules les fonctions définies ci-dessus sont acceptées. Donner la complexité algorithmique de la fonction implémentée. Vérifier le bon comportement de vos fonctions.**

**Question 3** Implémenter une fonction `simulation_pile` prenant comme arguments la hau-

teur de la pile et le nombre de grains à faire tomber. Cette fonction devra retourner une liste de piles de grain de sables. Cette liste stockera donc les différents états de la pile au fur et à mesure de la chute des grains.

**Question 4** Utiliser la fonction `trace_ecoulement` pour tracer l'écoulement des grains sur une pile.

**Question 5** Redéfinir la fonction `empiler` en la nommant `empilerSable` pour que le seul élément empilable soit la chaîne de caractères `"*"`. Ainsi, une pile de sable sera constituée d'une pile d'étoiles.

## Gestion du tas de sable

**Objectif** L'objectif est de modéliser les fonctions élémentaires afin de pouvoir manipuler un tas de sable et pas une colonne de sable.

Un tas de sable est maintenant modélisé par une **liste de piles** de grains de sable appelée `sablier`.

**Question 6** Implémenter la fonction `creation_sablier` d'arguments `larg` et `haut` permettant de créer un sablier constitué de `larg` piles de hauteur `haut`.

**Question 7** Modifier les fonctions `empiler` et `depiler` de telles sortes qu'elles prennent comme argument un `sablier` et un numéro de colonne `col`. On pourra ainsi déposer ou supprimer un grain de sable sur une colonne `col`.

## Modélisation de la chute libre d'un grain

**Objectif** On souhaite commencer par simuler la chute libre d'un seul grain.

Pour observer la chute d'un grain, on stockera dans une variable `simu` les différents états de la variable `sablier`. Ainsi `simu` est une liste de `sablier`.

**Question 8** Implémenter la fonction `chute_libre_grain` d'arguments `sablier`, `col` (numéro de la colonne ou le grain chute) et `simu`. Cette fonc-

tion retourne la variable `simu`.

**Question 9** Visualiser la chute d'un grain en utilisant la fonction `trace_sablier`.

### Modélisation de la chute d'un grain

**Objectif** Dans un premier temps on considère que le grain tombe toujours d'un même côté (à gauche par exemple).

On souhaite simuler ce comportement.

Le comportement attendu est le suivant :

- le premier grain tombe et reste dans la même colonne;
- le second grain tombe dans la même colonne puis tombe dans la colonne de gauche;
- le troisième grain tombe dans. Il ne peut pas tomber à gauche puisque la place est prise. Il reste au même endroit;
- le quatrième grain tombe dans la première colonne, puis tombe dans la colonne immédiatement à gauche, puis tombe une seconde fois;
- ...

**Question 10** Implémenter la fonction `chute_grain` d'arguments `sablier`, `col` et `simu`. Cette

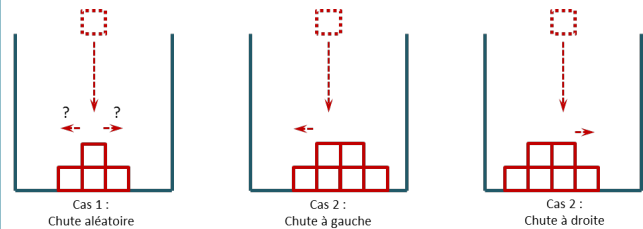
fonction retourne la variable `simu`.

**Question 11** Implémenter la fonction `simulation` d'arguments `nbgrains`, `larg` et `haut`. Cette fonction permet de déterminer tous les états du sablier permettant de simuler la chute de `nbgrains` dans un sablier de largeur `larg` et de hauteur `haut`. Cette fonction retourne la variable `simu`.

**Question 12** Visualiser la chute de 10 grains en utilisant la fonction `trace_sablier`.

### Synthèse

On identifie les 3 cas suivants pour déterminer le sens de chute d'un grain :



**Question 13** Simuler et visualiser la chute de 10 grains après avoir modifié la fonction `chute_grain` pour que les grains puissent chuter à gauche ou à droite.