

Algorithmique

CC 1 2022 / 2023

v.pagé

Tactique (15 pts)

Exo 1 (6pts)

Calcul d'une moyenne pondérée. On dispose des notes d'un étudiants dans un tableau nommé **notes**. On dispose par ailleurs des coefficients de chaque matiere dans un autre tableau nommé **coeff**.

On cherche à calculer la moyenne de cet etudiant, chaque matière étant pondérée par son coefficient. (*Le coefficient d'une matière dans la ième case du tableau de note est indiqué dans la ième case du tableau de coefficient*)

On pourra prendre par exemple :

```
notes = [12, 14, 8, 17]
coeff = [2, 1, 6, 1]
```

- (2 pts) Donner le code permettant de calculer la somme des coefficients (dans l'exemple : 10). *N'utilisez pas la constante 4 comme nombre de cases du tableau, utilisez la fonction len pour la calculer*
- (4 pts) Donner le code permettant de calculer la moyenne des notes. *Ici encore, n'utilisez pas la constante 4 comme nombre de cases du tableau, utilisez la fonction len pour la calculer*

Exo 2 (9 pts)

Le prix d'un bien immobilier évolue, d'une année sur l'autre comme suit :

- il gagne 10% de sa valeur courante (s'il valait 10000 euros, il en vaudrait 11000 la première année, puis 12100 la seconde) du fait de l'évolution des prix du marché
- Il se déprécie de 800 euros par an à cause de sa vétusté.

Au total, s'il valait 10000 euros au départ, il vaudrait donc 10200 euros l'année suivante

- (3 pts) Faire une fonction qui permet de calculer son prix l'an prochain du bien, connaissant son prix actuel
- (3 pts) Donner le code permettant de fixer le prix initial d'un bien (10 000 euros), fixer un nombre d'années (10), puis calculer le prix du bien au bout de ces années. On n'affichera que le prix final.
- (3 pts) Donner le code permettant de fixer le prix initial d'un bien (10 000 euros) et de calculer le nombre d'années nécessaires pour que ce prix double. On n'affichera uniquement ce nombre d'années

Strategie (5 pts)

On dispose d'un robot assez évolué disposant d'un GPS, capable de regarder son environnement (un champ agricole), de se déplacer, et de planter des graines.

Pour le programmer, on dispose des fonctions suivantes :

- **verifier_stock** : aucun argument. Renvoie le nombre de graines (entier) en stock dans le robot.
- **regarder** : sans argument. Renvoie une variable contenant des informations sur le champ. Il suffit de le faire une seule fois dans une session de travail du robot.
- **calculer_position** : sans argument. renvoie une variable contenant la position GPS du robot.
- **calculerProchaineAction** : prend comme arguments les informations sur le champ et la position GPS du robot. Renvoie une action qui est une chaîne de caractère : "avancer", "droite", "gauche", "planter", "fin"
- **deplacer** : prend comme argument une chaîne de caractères ("avancer", "droite" ou "gauche") ne renvoie rien mais déplace le robot selon cette commande

- **planter** : aucun argument. plante une graine à la position actuelle.

on suppose notre robot placé quelque part dans le champ, avec un stock de graine quelconque.

Faire un programme permettant de déplacer le robot automatiquement, jusqu'à ce que le stock de graine soit vide ou que le robot ait décidé qu'il avait terminé le traitement du champ (auquel cas l'action calculée est "fin"). Il devra alors afficher le nombre de graines plantées depuis le début du programme.

Si c'est trop difficile, essayez déjà de faire un bout de programme qui essaye de calculer simplement la prochaine action à réaliser, puis plante une graine si la commande est "planter". *ca vaudrait 2 pts*