

Tutorat de Programmation

L1 Informatique

Vous allez réaliser un programme qui permet de calculer les notes obtenues par des étudiants de L1 au CERI lors de leur dernier contrôle de programmation sur papier.

Les notes sont calculées à partir de fichiers dont la structure est fixée et dont voici un exemple :

1 Chuck Norris	Les lignes 1 à 8 sont un préfixe contenant le nom de l'étudiant et permettant au membre de l'équipe pédagogique chargé de l'évaluation de se souvenir des points à prendre en compte.		
2			
3 H - HEADER			
4 R - RETURN			
5 S - SYNTAX			
6 AI - ALGO IDEA			
7 AC - ALGO CORRECT			
8			
9 1 H 0.6 1	Cet examen avait 6 questions. Le barème fournit par le responsable de l'UE définit le nombre de point pour chaque question (oui c'est noté sur 22) :		
10 1 R 0.6 1			
11 1 S 0.6 0			
12 1 AI 0.6 1			
13 1 AC 0.6 1			
14			
15 2 H 0.8 1	Q1 → 3 pt		
16 2 R 0.8 1	Q2 → 4 pt		
17 2 S 0.8 1	Q3 → 4 pt		
18 2 AI 0.8 0.5	Q4 → 4 pt		
19 2 AC 0.8 0	Q5a → 4 pt		
20	Q5b → 3 pt		
21 3 H 0.8 1			
22 3 R 0.8 1			
23 3 S 0.8 1	La première colonne correspond à l'identifiant de la question.		
24 3 AI 0.8 0.5			
25 3 AC 0.8 0			
26			
27 4 H 0.8 1	Pour chaque question, le nombre de points est divisé également selon 5 critères : H, R, S, AI et AC. Le critère pour chaque ligne est situé dans la seconde colonne.		
28 4 R 0.8 1			
29 4 S 0.8 1			
30 4 AI 0.8 1			
31 4 AC 0.8 1			
32			
33 5a H 0.8 1	La troisième colonne correspond au nombre de points possible de gagner pour ce critère. (barème de la question / nombre de critères)		
34 5a R 0.8 1			
35 5a S 0.8 1			
36 5a AI 0.8 1			
37 5a AC 0.8 1			
38			
39 5b H 0.6 1	La quatrième colonne est remplie manuellement lors de l'évaluation et correspond au pourcentage de points à attribuer à l'étudiant pour chaque critère :		
40 5b R 0.6 1			
41 5b S 0.6 1			
42 5b AI 0.6 0			
43 5b AC 0.6 0	Ligne 9 : l'enseignant a attribué 100 % de 0.6 points.		
44	Ligne 18 : l'enseignant a attribué 50 % de 0.8 points.		

I – Première version

Ouvrez le fichier fourni **grade.py** et implémentez les fonctions dont les header vous sont fournis.

La fonction **compute_line_points** prends comme argument une chaîne de caractères correspondant à une des lignes du fichier (après la ligne 8) et calcule les points correspondants. Si la ligne est vide, la fonction doit renvoyer 0.

La fonction **sum_line_points** prends comme argument une liste de chaînes de caractères (liste de lignes), et calcul le total des points représentés par cette liste.

La fonction **compute_grade** prends comme argument une chaîne de caractères représentant un nom de fichier de notation, l'ouvre et calcul la note de l'étudiant.

Chaque fonction doit utiliser la fonction précédente.

II – Seconde version

Dans le fichier **compute_grades.py**, importez le module grade. Ce programme attends comme argument une liste de fichier, calcule et affiche la note pour chacun des fichiers ainsi que la moyenne.