

Sujet K2.A

On veut vérifier si vrai ou faux une chaîne de caractère non nulle est bien parenthésée. Proposez des tests unitaires couvrant les scénarios typiques, ainsi que le cas où la chaîne de caractères est vide qui est bien parenthésée.

Exemple: l'entrée “()” est bien parenthésée, mais “())” ne l'est pas.

Sujet K2.B

On veut vérifier si vrai ou faux, une chaîne de caractères non nulle contenant un ensemble de parenthèses, accolades, et de crochets est bien fermée. Proposez des tests unitaires couvrant les scénarios typiques, ainsi que le cas où la chaîne de caractères est vide, qui elle est bien fermée.

Exemple: l'entrée “[{}]” est bien fermée, mais “[{}])” ne l'est pas.

Sujet K2.C

On veut vérifier si vrai ou faux, une chaîne de caractères non nulle a un espacement valide. Pour ce kata, la définition d'un espacement valide est : un seul espace entre les mots, sans espace(s) avant ou arrière. S'il n'y a pas d'espace tout court dans la chaîne de caractères, elle a un espacement valide.

Proposez des tests unitaires couvrant les scénarios typiques décrits par ce sujet. Par exemple, l'entrée “ test espace !” n'a pas un espacement valide, mais “test” et “hello world” si.

Sujet K2.D

Certaines personnes n'ont qu'un prénom, d'autres ont un prénom et un nom de famille, et certaines ont un prénom, plusieurs prénoms secondaires et un nom de famille. Votre ami développer doit écrire fonction qui initialise les prénoms secondaires, c'est-à-dire que chaque prénom secondaire doit être remplacé par sa première lettre suivie d'un point (.)

Proposez lui des tests unitaires couvrant les scénarios typiques.

Exemple : l'entrée "Noromino Helena Nirina Randrianaly" donne "Noromino H. N. Randrianaly" tandis que l'entrée "Noromino Randrianaly" reste "Noromino Randrianaly".

Sujet K2.E

On veut supprimer les doublons d'une liste d'entiers, en conservant la dernière occurrence de chaque élément (la plus à droite). Par exemple, pour l'entrée : [3, 4, 4, 3, 6, 3], il faut supprimer le 3 à l'index 0, supprimer le 4 à l'index 1 et supprimer le 3 à l'index 3. Le tableau sera toujours non nul, mais peut potentiellement être vide.

Proposez des tests unitaires pour couvrir les scénarios typiques.

Sujet K2.F

On veut savoir quelle heure il sera dans une minute. Pour cela on aura une fonction qui prend en compte deux paramètres : le nombre d'heures et de minutes, et qui va retourner un tableau qui contient deux éléments : 1) le nombre d'heures après une minute, 2) le nombre de minutes après une minute.

Proposez des tests unitaires pour couvrir les scénarios possibles.

Sujet K2.G

On veut une fonction qui prend une chaîne de caractères non-nul, et un entier $n > 0$ comme paramètres et qui renvoie une liste de tous les mots dont la longueur est supérieure à n .

Exemple : l'argument "The quick brown fox jumps over the lazy dog", 4 retourne ['quick', 'brown', 'jumps']

Écrire les tests unitaires pour couvrir les cas typiques qui découlent de cet énoncé.

Sujet K2.H

On veut une fonction qui filtre un tableau de noms pour rechercher un mot-clé sans tenir compte de la casse. Par exemple, si parmi le tableau pris comme premier argument, contenant Jean-Pierre, Jeannot, Jacques, on voulait filtrer ceux qui ont "jean" on devrait avoir Jean-Pierre, Jeannot dans le tableau résultant.

Écrire les tests unitaires pour couvrir les scénarios typiques de cet exercice.

Sujet K2.I

Vous avez deux tableaux : le premier contient la clé des bonnes réponses à un examen, et le second les réponses d'un étudiant. Les deux tableaux ont la même longueur. La fonction doit retourner le score de l'étudiant en attribuant : +4 pour chaque bonne réponse, -1 pour chaque mauvaise réponse, +0 pour chaque réponse vide (chaîne vide).

Si le score est inférieur à 0, retournez 0. Écrire les tests unitaires pour couvrir les scénarios possibles de cet exercice.

Exemple : pour les arguments ["a", "a", "b", "b"] et ["a", "c", "b", "d"] l'étudiant devrait avoir 6 points, puisque sur les 4 questions il obtient $4 + (-1) + 4 + (-1)$ points, soit 6 points en tout.

Sujet K2.J

On voudrait s'assurer qu'un tableau de chaîne de caractères non-nul, est trié dans le sens souhaité (ascendant / descendant). La fonction aura donc deux paramètres : le tableau à trier, et le sens du tri qui peut être "ASC" ou "DESC". Dans le cas où le sens n'est ni ASC ni DESC, on retourne le tableau original sans modification. Après chaque appel de cette fonction, le tableau original ne doit pas être modifié.

Proposez des tests unitaires pour couvrir les scénarios de cet énoncé.

Sujet K2.K

On souhaite soustraire deux tableaux non-nuls, composés de chiffres, potentiellement vides. C'est-à-dire que pour un tableau A, on va retirer tous les éléments de B qui y sont présents, et on va retourner le reste dans un tableau.

Par exemple, si A contient les chiffres [1,2,2] et B contient [2, 3], on aurait en retour le tableau [1].

Proposez des tests unitaires pour couvrir les scénarios possibles qui découlent de cet énoncé.

Sujet K2.L

Dans une ligne téléphonique on souhaite connaître le nombre de personnes maximum qui étaient dans la même cabine aujourd'hui. Pour cela, on va créer une fonction dont le premier paramètre est le nombre de personnes qui descendent à chaque arrêt, et le deuxième est le nombre qui montent à cet arrêt. Ces deux tableaux auront toujours la même longueur.

Par exemple, pour les arguments $[0, 2, 4, 4]$ et $[3, 5, 2, 0]$, on aura la réponse 6 :

- Le nombre de passagers dans le tram au début est de 0.
- À la première station, 3 passagers montent dans la cabine, et le nombre de passagers dans la cabine devient 3.
- À la deuxième station, 2 passagers descendent de la cabine (1 passager reste à l'intérieur). Puis 5 passagers montent dans la cabine. Il y a maintenant 6 passagers dans la cabine.
- À la troisième station, 4 passagers descendent de la cabine (2 passagers restent à l'intérieur). Puis 2 passagers montent dans la cabine. Il y a maintenant 4 passagers dans la cabine.
- Enfin, tous les passagers restants dans la cabine descendent lors de la dernière station. Il n'y a plus de passagers qui montent dans la cabine.

Proposez des tests unitaires couvrant 3 scénarios possibles en dehors de l'exemple cité par le sujet. Attention, vos données de test doivent respecter cette contrainte : la cabine est toujours vide au premier arrêt, et tous les passagers restants descendent au dernier arrêt, et personne ne monte au dernier arrêt car c'est le terminus.