

Projet 1e : Test d'une carte de crédit (992e-3787836)

Pour savoir si un numéro de carte de crédit est valide, on utilise très souvent l'algorithme de Luhn 4. Ce numéro est un entier composé de 16 chiffres. Le dernier chiffre est la clé qui permet de contrôler l'exactitude du numéro.

Le principe de l'algorithme de Luhn est le suivant. On commence toujours par le chiffre se trouvant le plus à droite. Ce chiffre sera le premier élément de la liste dites des "indices impairs". Puis on complète cette liste en prenant un chiffre sur deux du numéro de carte bancaire, toujours en le lisant de la droite vers la gauche. Pour la liste des chiffres "d'indices pairs", on commence par le deuxième chiffre le plus à droite du numéro de la carte de crédit, on se déplace de la droite vers la gauche comme pour la liste précédente et on construit la liste, en prenant un chiffre sur deux. Pour les nombres de cette liste des indices pairs, on double tous les chiffres. Si un nombre est supérieur à 9, on réalise la somme des deux chiffres qui le composent (exemple si on obtient 16, on additionne 1 et 6 pour avoir 7). Par conséquent, tous les nombres des deux listes sont composés uniquement de chiffres compris entre 0 et 9. On calcule alors la somme totale des chiffres de ces deux listes. Si cette somme est un multiple de 10, alors le numéro de la carte de crédit est valide.

- 1) Écrire une fonction `num_en_liste` qui transforme un nombre entier en une liste de chiffres. Cette fonction a un paramètre de type `int` et retourne un élément de type `list`.

exemple : `num_en_liste(12345)` renvoie `[1,2,3,4,5]`

- 2) Écrire une fonction `tuple_pairs_impairs` qui détermine un tuple représentant la liste des chiffres d'indice pair et la liste des chiffres d'indice impair d'un numéro de carte de crédit. Le chiffre le plus à droite de ce numéro est considéré comme le premier chiffre d'indice impair. Cette fonction a un paramètre de type `int` et retourne un tuple composé de deux éléments de type `list`.

#exemple : `tuple_pairs_impairs(12345)`=>(`[4, 2]`, `[5, 3, 1]`)

- 3) Écrire une fonction `traitement_nb_pairs` qui multiplie par 2 tous les chiffres de la liste passée en argument. Si un chiffre obtenu est supérieur à 9, il faut le remplacer par la somme des deux chiffres qui le composent. Cette fonction a un paramètre de type `list` et retourne une liste.

#exemple : `traitement_nb_pairs([5,6,7])`=>`[1,3,5]`

- 4) Utiliser les fonctions précédentes afin de tester la validité de votre carte de crédit