

# Tutorat de Programmation

## L1 Informatique

### I – Retours sur le CC1

Utilisez le fichier fourni cc1.py pour implémenter les fonctions du premier contrôle sur feuille. Les chaînes de documentation de chaque fonctions comportent les tests qui seront lancés automatiquement lors de l'exécution du fichier.

Écrivez vos solutions directement dans le fichier.

De manière générale, lorsque vous devez passer ce genre d'examen, écrivez le minimum nécessaire demandé. Ici, était demandé d'écrire uniquement des fonctions. Tout code en dehors d'une fonction est une perte de temps.

Pour chaque écriture de fonction, vous devez tout le temps vous poser les mêmes questions

- Quel Nom donner à la fonction, et quels sont les paramètres nécessaires ?
- Est-ce que la fonction doit afficher quelque chose ? (Si pas explicitement demandé : jamais)
- Quel est le type des paramètres, quel est le type de la valeur renvoyée par la fonction ?
- Une fonction peut renvoyer None mais avoir un effet sur ses paramètres, est-ce le cas ?

#### Question 1 :

Nom : split. Paramètre : l'entier à couper.

Affichage : Hell No.

Type : paramètre : entier – valeur renvoyée : tuple d'entiers

Effet sur les paramètres : non

#### Question 3 :

Nom : remove\_odd\_numbers, Paramètre : la liste à traiter

Affichage : Fuck No.

Type : paramètre : liste d'entiers, valeur renvoyée : None

Effet sur le paramètre : oui

#### NB :

- Pour la question 2, l'énoncé était ambigu et on pouvait supposer ou pas que la fonction pgcd() existait déjà et pouvait être utilisée directement.
- Pour la question 3, la plus part des étudiants ont eu une idée de l'algo à implémenter, mais presque personne ne l'a fait correctement. (On ne peut pas utiliser un simple for i in range())

## II – No if required

```

3 def isZero(x):
4     if x == 0:
5         return True
6     else:
7         return False

```

Cette fonction prends un entier, vérifie si c'est zéro, renvoie True si c'est le cas et sinon renvoie False. Considérez ce qu'il se passe lors de l'exécution de cette fonction dans les deux cas suivants :

La fonction est invoquée avec la valeur 2 pour argument.

On se retrouve à l'intérieur de la fonction, ligne 4 et x vaut 2. L'interpréteur Python se retrouve face à une expression « if ». Une expression if est toujours exécutée de la même manière : la condition est évaluée. Si le résultat de l'évaluation vaut « True » alors la branche « if » est exécutée. Sinon la branche « else » est exécutée :

```

1     if 2 == 0:
2         return True
3     else:
4         return False

```

... `2 == 0` est évalué, et comme dans le shell, produit la valeur « False »

```

1     if False:
2         return True
3     else:
4         return False

```

... Puisque la condition vaut « False », la branche « else » est exécutée et « False » est renvoyé.

Maintenant si on appelle la fonction 0 comme argument, l'exécution correspondante est :

```

1     if 0 == 0:
2         return True
3     else:
4         return False

```

... `0 == 0` est évalué, et comme dans le shell, produit la valeur « True »

```

1     if True:
2         return True
3     else:
4         return False

```

... Puisque la condition vaut « True », la branche « if » est exécutée et « True » est renvoyé.

Remarquez que dans les deux cas, l'évaluation de la condition produit la même valeur que celle qui finit par être renvoyée. Vous pouvez donc renvoyer l'expression `x == 0` (autrement dit, la condition) directement.

Faites les exos dans le dossier noif. Utilisez une seule ligne de code par fonction.