

## Second projet du semestre 1 pour le cours de M2 "Calculabilité"

**Date de rendu:**15/01

**Mode de rendu:**Archive électronique à déposer sur une adresse de téléversement:

Mode de participation: **individuel ou en binome.**

### Question I

L'**objectif** est d'écrire en python un traducteur de programmes MTdV vers un sous-ensemble de Python de manière à ce que l'exécution du programme MTdV et votre traduction en Python s'exécute de manière identique.

Le sous-ensemble de Python considéré dans cette question est composé :

- 1- des constantes numériques entières,
  - 2- des opérations en arithmétique entière : l'addition et la soustraction,
  - 3- des variables globales entières individuelles (autant que vous voulez),
  - 4- d'une unique variable globale de type "list",
  - 5- de l'instruction d'affectation,
  - 6- des instructions d'affichage print() et format()
  - 7- de l'instruction de test "if else: ..."
  - 8- des valeurs des arguments initiaux passés par la commande shell ("\_\_main\_\_" , les module "sys", "argv") représentés par n+1 variables : ARG0, ARG1,...,ARGn pour les arguments et ARGC pour le nombre d'arguments (n).
  - 9- et de procédures Python (l'instruction "return" est prohibée) n'ayant aucun paramètre, ni variables locales.
- Toute autre type d'instruction dans le programme Python généré est interdite.

Les **livrables** attendus sont:

- a) l'explicitation de la traduction de chaque instruction MTdV vers le code Python,
- b) du code Python de votre traducteur qui doit être capable de fonctionner avec tous les programmes MTdV fournis en cours.

### Question II

L'objectif de cette question est de refaire le travail de la question 1 en levant certaines contraintes, vous aurez maintenant droit à un sous-ensemble plus large d'instruction Python comme cible de votre traduction qui est composé :

- 1- des constantes numériques entières,
- 2- des opérations en arithmétique entière : l'addition et la soustraction,
- 3- des variables globales entières individuelles (autant que vous voulez),
- 4- d'une unique variable globale de type "list",
- 5- de l'instruction d'affectation,
- 6- des instructions d'affichage print() et format()
- 7- de l'instruction de test "if else: ..."
- 8- les arguments initiaux passés par la commande shell ("\_\_main\_\_" , les module "sys", "argv") représentés par n+1 variables : ARG0, ARG1,...,ARGn pour les arguments et ARGC pour le nombre d'arguments (n).

**10**<sup>1</sup>- de fonctions Python pouvant avoir de 0 à n paramètres et des variables locales et bien sûr au moins une instruction "return".  
Toute autre instruction dans le programme Python généré est interdite.

Les livrables attendus sont:

- l'explicitation de la traduction de chaque instruction MTdV vers le code Python
- du code Python de votre traducteur qui doit être capable de fonctionner avec tous les programmes MTdV fournis en cours.

### Question III

L'objectif de cette question est de refaire le travail de la question 2 en ajoutant les contraintes : *"l'instruction d'affectation est interdite" ainsi que "l'absence de variable locale dans le corps des fonctions"*<sup>2</sup>. Votre sous-langage Python cible est donc composé :

- 1- des constantes numériques entières,
- 2- des opérations en arithmétique entière : l'addition et la soustraction,
- 11- les variables globales sont interdites à l'exception des variables représentant les arguments d'initialisation comme décrits dans la contrainte 8 précédemment, c-à-d. "8- les arguments initiaux passés par la commande shell ("\_\_main\_\_" , les module "sys", "argv") représentés par n+1 variables : ARG0, ARG1,...,ARGn pour les arguments et ARGC pour le nombre d'arguments (n)",
- 6- des instructions d'affichage print() et format()
- 7- de l'instruction de test "if else: ..."
- 12- de fonctions Python pouvant avoir de 0 à n paramètres, au moins une instruction "return", mais pas de variable locale,  
Toute autre instruction dans le programme Python généré est interdite.

Les livrables attendus sont:

- l'explicitation de la traduction de chaque instruction MTdV vers le code Python
- du code Python de votre traducteur qui doit être capable de fonctionner avec tous les programmes MTdV fournis en cours.

### Question IV (optionnelle)

L'objectif de cette question est de refaire le travail de la question 3 en ajoutant une seule contrainte: ***"les fonctions ne peuvent avoir au plus qu'un unique argument"***.

Les livrables attendus sont:

- l'explicitation de la traduction de chaque instruction MTdV vers le code Python
- du code Python de votre traducteur qui doit être capable de fonctionner avec tous les programmes MTdV fournis en cours.

---

1 La numérotation des contraintes définissant l'ensemble des instructions Python autorisées dans la traduction d'un programme MTdV est globale au sujet du projet, de façon à ce qu'un même numéro ne soit pas attribué à deux contraintes différentes. En conséquence, il y aura des discontinuités dans la numérotation de certains groupes de contraintes.

2 Mais les fonctions peuvent avoir des paramètres.