

Un TP de FOLie

On est jamais vraiment sûr de rien.

A. Introduction

Pour changer les habitudes, ce TP sera entièrement à faire dans un nouveau langage : **FOLie**

A la différence de **C** ou **OCaml** qui sont des langages compilés, **FOLie** est un langage interprété (comme python). Un compilateur peut se voir comme un traducteur qui traduit une seule fois le code voulu (souvent vers le langage machine), en revanche un interpréteur lit le code et exécute la tâche appropriée en même temps.

L'extension des fichiers de code est « `· ·` », par exemple un fichier de code en **FOLie** se nommera de la manière suivante : `NOM_DU_FICHIER...`

Pour exécuter un de vos fichiers, lancez l'exécutable **FOLie** en terminal puis lorsqu'il vous demande d'entrer un fichier donnez lui le chemin depuis l'endroit où se situe l'exécutable.

I. Explications

Ce langage a un fonctionnement particulier qui n'a rien à voir avec **C** ou **OCaml**, nous allons donc voir ici comment l'utiliser¹ et coder dans ce langage.

- * Le langage se base entièrement sur un système de **PILE** (structure de type **FOLI**² ou presque), chaque opérateur et fonction utilise les éléments de la pile pour s'exécuter, et donc il n'y a pas de variables, uniquement des éléments à différents endroits de la pile. La pile ne peut contenir que des entiers et des booléens.

Définition 1 On définit la relation \trianglelefteq telle que :

$a \trianglelefteq b$ si a est plus **profond** que b dans la pile ou que a est b .

$a \trianglelefteq b$ se lira « b assiège a ».

- * Chaque ligne d'un programme correspond à une fonction et le nom de cette fonction est le numéro de la ligne sur laquelle elle a été écrite, un code correspond donc à une suite de fonctions écrites les unes à la suite des autres.
- * Sur chaque ligne il est possible d'écrire :
 1. des **opérateurs** comme $+$, $-$, \times et d'autres qui seront définis plus tard.
 2. des suites de lettres finissant par une majuscule et n'ayant aucune autre majuscule, cela correspond à des **mots clés** qui auront certaines fonctionnalités définies plus tard. Par exemple «entier» est une syntaxe valide pour un mot clé, «enTier» ou «Entier» ne le sont pas
 3. des suites de lettres n'ayant une majuscule ni au début ni à la fin, cela correspond à des **commentaires**. Par exemple «bonjour» et «meRci» sont des commentaires valides, «Bonjour» et «meRcI» ne sont pas des commentaires valides.
 4. des suites de lettres commençant par une majuscule et n'ayant aucune autre majuscule, cela correspond à des **nombres**. Par exemple «Beg» est une syntaxe valide de nombre, «bEg» lui n'est pas valide (et **attention** «12» n'est pas une syntaxe valide de nombre non plus). Mais cependant ces nombres doivent être écrits dans une certaine base, pour la ligne i les nombres sont en base $5 + (i \bmod 10)$, et donc en

¹En partie, vous découvrirez certaines spécificités par vous même

²FOLI pour First Out Last In

base i on a accès uniquement au i premières lettres de l'alphabet pour écrire nos «nombres». Dès qu'un de ces «nombre» est écrit sur une ligne il est simplement ajouté au dessus de la pile. Petit exemple : sur la ligne 42 nous somme donc en base 7, donc Beg correspond à 83 sur cette ligne. Par exemple «A B C» est une ligne valide, elle ajoute simplement 0, 1 et 2 dans la pile (dans cet ordre).

- des nombres, ils correspondent à des **sauts conditionnels**. Plus explicitement, en fonction du booléen sur le dessus de la pile le programme exécute ou non la ligne (fonction) indiquée par le nombre écrit, puis à la fin de l'exécution de cette ligne, le programme reprends l'exécution la où il avait fais le saut conditionnel initialement. Par exemple si il est écrit sur une ligne : « 42 Bbb », en lisant 42 le programme exécutera la ligne 42 si le booléen sur le dessus de la pile le permet, puis reviendra à Bbb et continuera normalement, si le dessus de la pile n'est pas un booléen, l'appel à 42 renvoie une erreur.

Tous éléments différents doivent absolument être séparés par **au moins 1** espace.

- * Comme vous le savez probablement déjà, on est jamais sûr de rien. Et donc pourquoi nos choix de booléens serait uniquement vrai ou faux puisqu'il que l'on ne peut pas être entièrement sûr si quelque chose est vrai ou faux. Ainsi, on dispose de plusieurs booléens entre impossible et certain. On appellera booléen exact les booléens **certain** et **impossible**.

Exemple de ligne valide syntaxiquement: A essai 0 Ddd 0 fin

1. Que fait la ligne précédente ?
2. Écrire sur la ligne 3 le nombre 12.
3. Écrire sur la ligne 179 le nombre 87591074 (Bonus pour le lore).

II. Opérateurs et mots clés

Nous allons maintenant définir les opérateurs et les mots clés du langage.

Définition 2

*On appellera **arguments** d'une fonction les derniers éléments de la pile au début de son exécution.*

Voici les opérateurs :

- * **+** : prends en arguments deux entiers, fais leur somme et l'empile. Dépile les 2 arguments.
- * **-** : prends en arguments deux entiers, fais leur différence et l'empile (si a et b sont les 2 derniers éléments de la pile tels que $a \preceq b$, un appel à **-** empile $a - b$). Dépile les 2 arguments.
- * **×** : prends en arguments deux entiers, fais leur produit et l'empile. Dépile les 2 arguments.
- * **:** : prends en arguments deux entiers, fais leur division entière et l'empile. Dépile les 2 arguments.
- * **>** : prends en arguments deux entiers x et y tels que $x \preceq y$ et empile un booléen correspondant à une bonne approximation de $x > y$ et dépile x et y .
- * **?** : empile un booléen au hasard.
- * **_** : prends en argument un élément et le dépile (en gros ça supprime le haut de la pile).
- * **++** : prends en argument un élément et l'empile (en gros duplique le haut de la pile).
- * **&** : prends en arguments deux booléens x et y tels que $x \preceq y$ et empile x NAND y (NAND selon notre vision des booléens, à peu près la probabilité du complémentaire de l'intersection). Dépile les 2 arguments.
- * **<** : prends en arguments un entier n et un élément x tels que $x \preceq n$, et décale x de n dans la pile et dépile n .

Voici les mots clés :

- * **entierR** : demande un entier dans l'interpréteur et l'empile.
- * **booléenR** : demande un booléen dans l'interpréteur et en empile un selon votre réactivité.
- * **imprimeR** : prends en argument un entier, et affiche dans l'interpréteur le caractère ascii correspondant et le dépile, si l'argument est un booléen, il est affiché.
- * **afficheR** : prends en argument un entier ou un booléen, l'affiche dans l'interpréteur et le dépile.

Exemple de ligne 0 valide : ? entierR entierR + B < 0 imprimeR

4. Que fait la ligne précédente ?
5. (**Important**) Comment peut on rajouter sur le dessus de la pile le booléen **certain** à coup sûr (pour par exemple être sûr à 100 % d'effectuer un saut conditionnel).

III. Dernières définitions

Définition 3

On appelle **mauvaise fonction** une fonction qui termine avec une probabilité de 1 et dont les effets secondaires sont proches de ceux qui sont attendus.

Définition 4 : (RAA)

On appelle **mauvaise mauvaise fonction** une fonction totalement correcte.

B. Exemples

Le but de cette partie est de vous faire prendre connaissance du langage en implémentant des fonctions simples et qui vous seront utiles tout au long du TP.

6. Écrire une mauvaise mauvaise fonction qui affiche "Hello World !" ³⁴
7. Écrire une mauvaise mauvaise fonction qui prends 2 arguments a et b en entrée et les remplace par le reste de la division euclidienne de a par b (où $a \preceq b$).
8. Montrer que \preceq est une relation d'ordre totale sur les éléments de la pile.
9. Écrire une mauvaise mauvaise fonction qui prends en arguments a et b 2 entiers et met en haut de la pile le booléen exact $a = b$.

C. Le vif du sujet

Maintenant que vous êtes échauffés vous allez pouvoir attaquer les vrais exercices du TP :

La suite du TP se déroule ainsi : vous disposez d'un fichier nommé « **Exercice1Etoile.pdf** », dedans se trouvent divers exercices (fonctions à coder), essayez vous à celui que vous voulez (pas besoin de les faire dans l'ordre) et lorsque vous avez fini au moins un des exercices de ce fichier, vous avez deux choix, soit continuer avec les autres exercices du fichier, soit m'appeler (moi obligatoirement) pour que je vous donne un deuxième fichier nommé cette fois ci « **Exercice2Etoile.pdf** » où se trouvent des exercices légèrement plus difficiles. Je dispose au total de 4 fichiers supplémentaires et j'en donnerais l'accès uniquement si un exercice d'un niveau précédent à été implémenté convenablement.

³"Halo an t-saoghail"

⁴Vous pouvez aussi afficher $\pi\rho\tau\alpha\pi\rho\iota\lambda\iota\alpha$ en grec à la place

D. Bingo ! (Bonus)

Voici une grille de **Bingo** liée aux affichages de l'interpréteur de **FOLiE**, libre à vous d'essayer de la remplir⁵ en rencontrant certains de ces affichages pendant le TP.

Il est long ce mot	$+\infty$	C est la goutte d'eau qui fait déborder le vase	On va où là	7 7 7 0 9 17 12 1 1 3 5 8 18 17 3 5 8 19 0 9 20 12 1 3 3 5 8 21 18 5 9 17 0 27 14 1 3 5 9 20 15
Me coupe pas la parole	C est dur à calculer	Mais qui est ce rustre	Je n apprécie pas ce que je vois	STOP
Pffffff.	$0 \notin \mathbb{R}^*$	Il est où ?	?	Je suis censé deviner
Je suis l'interprète, la voix du code dans sa forme la plus pure et la plus chaotique.	Je ne peut pas aller à une ligne qui n'existe pas	«Tous les animaux sont égaux, mais certains sont plus égaux que d'autres.» - Georges Orwell, <i>La ferme des animaux</i> , 1945	Et c'est moi qui choisis ?	Ça ne fait pas QUE afficher 20, s21 c'est bien plus que ça
Je n apprécie pas ce que je vois	Tu connais la différence entre un caractère et une chaîne ?	La fonction ... n'existe pas	Attends je cherche	Là c'est ta faute

⁵Si vous y arrivez appelez moi, d'ailleurs si vous obtenez la citation de Georges Orwell ou la suite de nombres séparés par des barres verticales ou l'interprète fou, appelez nous qu'on rigole.