

Projet INF404

SELME Kelyan

TERESE Niels

1. Introduction

Pour ce projet, nous sommes partis sur la réalisation d'un petit interpréteur.

Après compilation, le programme de l'interpréteur lit en entrée un fichier et interprète ce fichier

2. Fonctionnalités

Notre interpréteur sait reconnaître et interpréter les commandes suivantes :

- `lire(X);` lit au clavier un nombre et l'affecte à une variable (ici *X*)
- `ecrire(X);` affiche à l'écran une variable (ici *X*)
- `X = aff;` affecte une valeur *aff* à une variable (ici *X*). *aff* peut être un nombre, une variable, ou une opération mathématique incluant des nombres et/ou des variables.
- `if condition then action1 else action2 fi;` interprète *action1* si *condition* est vraie et interprète *action2* sinon. *condition* est sous la forme d'une comparaison (`==`, `!=`, `<=`, `<`, `>`, `>=`) entre 2 nombres / variables. Une condition peut comporter plusieurs conditions séparées par `||` ou `&&`
- `while condition do action done;` interprète *action* tant que la condition *condition* est vraie. *condition* est sous la même forme que la

3. Exemples

Voici différents exemples qui couvrent l'ensemble des fonctionnalités de notre interpréteur :

```
lire (X);
Y = X + 1;
ecrire (Y * 2);
```

Ce programme demande une entrée au clavier, et affiche une retour au terminal

```
a = 1;  
b = 2;  
c = b + 3;
```

Ce programme demande gère diverses affectations de variables

```
lire(x);  
lire(y);  
if x <= 10 && y <= 2 then  
    x = x + 1;  
else  
    y = y + 1;  
fi;  
ecrire(x);
```

Ce programme demande une entrée au clavier, et agit en fonction de 2 conditions

```
y = 5;  
while y < 10 do  
    y = y + 1;  
done;
```

Ce programme demande effectue une action dans le cadre d'une boucle while