MINI PROJET

Conception d'un mini compilateur pour le langage « FROG »

Le but du mini projet est de développer un compilateur capable d'analyser (**lexicalement, syntaxiquement** et **sémantiquement**) un fichier source **(.FRG)** écrit sous un langage nommé **« FROG».**

Dans l'analyse lexicale toutes les unités lexicales (tokens) doivent être reconnues.

Dans l'analyse syntaxique, les erreurs syntaxiques doivent être signalées dans le cas ou une ou plusieurs instructions sont mal écrites.

Dans l'analyse sémantique, les erreurs sémantiques doivent être affichées telles que l'incompatibilité entre les types, l'utilisation de variables non déclarées, etc.…

Spécification du langage « FROG » :

Le code source doit être écrit de la manière suivante :

- Le programme commence toujours par **« FRG_Begin »** et se termine par **« FRG_End »**
- Une et une seule instruction par ligne.
- Une instruction se termine toujours par **« # »**
- Un commentaire prend une ligne entière et doit commencer par **« ##»**
- Un identificateur doit commencer obligatoirement par une lettre et peut être suivi par un caractère alphanumérique.
- Un nombre entier est composé au minimum par un chiffre (exemple : 40, 190, 7) et déclaré par le mot clé **« FRG_Int ».**
- Un nombre réel est composé de deux entiers séparés par un point exemple : (30.9, 19.01, 88.55) et déclaré par le mot clé **« FRG_Real ».**
- Une chaîne de caractère est mise entre deux quottes doubles et déclarée avec le mot clé **« FRG_Strg ».**

Les instructions à considérer dans ce langage sont illustrées dans l'exemple suivant :

exemple d'un fichier source	signification
FRG_Begin	début du programme
FRG_Int i, j, x1, x2 #	déclaration de 4 variables entières
FRG_Real x3 #	déclaration d'une variable réelle
i:=30 #	affectation d'une valeur entière à i
If [i<=20]	conditionnel
x1:=10 #	affectation de valeur entière à x1
Else	sinon
Begin	début de bloc
x1:=30 #	affectation d'une valeur entière à x1
x3:=x1*4 #	affectation d'une expression à x3
FRG_Print x1, x3 #	Affichage de la valeur de x1 et x3
End	fin de bloc
FRG_Print " Hello :) " #	Affichage d'un message
## instruction de boucle Repeat	commentaire
Repeat	boucle repeat
i:=i-5 #	affectation d'une expression
FRG_Print I #	affichage de la valeur de i
until [i <= 15]	condition de fin de la boucle
FRG_End	fin du programme

Le travail doit être présenté avec une interface graphique (comme illustrée ci-dessous) avec au moins un bouton pour charger un fichier source et trois autres boutons pour chaque type d'analyse. Ajouter un bouton pour afficher les résultats d'exécution du fichier source.

Le mini projet doit être donc développé avec le langage de programmation de votre choix (exemple : C, Java, python….)