**License 3 , ISIL Année Universitaire : 2019/2020, Module Compilation**

1. **Introduction**

Le but de ce projet est de réaliser un mini-compilateur d’un langage appelé « AlgoISIL», en effectuant les différentes phases de la compilation : lexicale, syntaxique, sémantique et générations de code intermédiaire.

1. **Description du Langage AlgoISIL**

**II.1 La structure générale**

PartieDeclarationBibliotheque

**ALGORITHME NOM**

**VAR** **/\* le programme peut contenir plusieurs fonctios**

\*/

**FONCTION NomFonction ( ListeParametre) :TYPE**

**VAR** PartieDeclarationDeVariable

**DEBUT**

PartieInstruction

**FIN**

PartieDeclarationDeVariable

**DEBUT**

PartieInstruction

**FIN**

**La partie de déclaration de bibliothèques**

* Le programme peut contenir une liste de bibliothèque qu’on doit déclarer au début de programme. Chaque bibliothèque **doit être sur une ligne séparée** et précédée par le symbole « $$».
* La liste des bibliothèques autorisées est :

|  |  |
| --- | --- |
| Nom | usage |
| $$include <InOut> | Pour les instructions ECRIRE et LIRE |
| $$include<Arithme> | Pour les affectations et les expression arithmétiques |

**La partie déclarations de fonctions**

Le langage peut contenir plusieurs fonctions, qu’on doit déclarer **dans la partie VAR**.

La déclaration d’une fonction suit la syntaxe suivante :

**FONCTION NomFonction ( ListeParametre) : TYPE**

**Tel que**

Les différents types d’une fonction sont les suivant :

|  |  |  |
| --- | --- | --- |
| Le type | Description | Exemple |
| entier | Un entier est une suite de chiffres. Signé ou non.Sa valeur est entre -32768 et 32767. | 5, -5 |
| reel | Une suite de chiffres contenant le point décimal. | 5.5, -5.5 |
| chaine | Elle représente une chaîne de caractères. Elle doit être placée entre guillemets. | **"**chaine de caractères de type**"** |

* **Non fonction** est un identificateur, qui est une suite alphanumérique qui commence par lettre ,et qui peut contenir un ‘\_’ ,et dont la taille ne doit pas dépasser 12 caractères.
* **ListeParamètre :** est une suite de paramètres séparés par « . ». Chaque paramètre est déclaré par la syntaxe

**NomVariable : TYPE**

**NomVariable** est un identificateur

**La Partie déclaration de variables**

Nous pouvons déclarer dans ce langage des variables (simples ou tableaux) et des constantes.

* Variable simple : nom variables : TYPE#
* Tableau : NomVariables [taille] :TYPE#
* Constante : **CONST** NomVariables : TYPE#

**taille** est un chiffre strictement positif.

La déclaration d’un ensemble de variables s’effectue en séparant les noms de variables par ‘.’. et chaque déclaration se termine par ‘# ‘

Exemple : NomVariable1. NomVariable2 : entier#

**La Partie instruction**

Dans notre langage, **SUELES** les instructions suivantes sont autorisées :

|  |  |  |
| --- | --- | --- |
| Instruction | Description | Exemple |
| Affectation | Idf :=expression arithmétique # | A :=2 #  A :=C+D #  Tab[2] :=a+b/2 # |
| Boucle | **tanque (** cond)  **Faire**  Bloc Instructions  **Fait** | tanque(i<j)  Faire  A :=2 #  A :=C+D #  Fait |
| ECRIRE | **ECRIRE(“chaine”);** | ECRIRE**(“hello”)#** |
| LIRE | **LIRE(listeVariable)** | LIRE(x.y) # |
| Nomfonction | **NomFonction(listeVariable)** | Fonct1(x.y)# |
| return | **Return Variable** | Return x# |

* **Les expressions arithmétiques** sont composées des opérateurs : +, \*,- ,/
* **Opérateurs de comparaison sont : =, < , >, <=, >=, !=**

**Associativité et Priorité des opérateurs**

* ***Associativité :*** gauche pour tous les opérateurs
* ***Priorité :*** les priorités sont données par la table suivante par ordre croissant :

|  |  |
| --- | --- |
| *Opérateurs Arithmétiques* | + - |
| \* / |

**II.9 Les commentaires**

Un commentaire peut être écrit sur une ou sur plusieurs lignes en mettant le texte entre « /\* » et « \*/ »

**Exemple** :

/\* ceci est

Un commentaire\*/

1. **Travail à réaliser :**

Ci-dessous les différentes phases à effectuer afin de réaliser le compilateur demandé.

* **Analyse lexicale avec l’outil FLEX.**
* **Analyse syntaxique avec l'outil Bison**
* **Analyse sémantique pour les erreurs suivantes :**
* Idf non déclaré
* Idf double déclarée
* Non compatibilité de type
* Dépassement de la taille d’un tableau
* Absence d’une bibliothèque nécessaire
* Modification d’une valeur d’une constante
* Vérification de fonctions : appelle d’une fonction déclarée, le respecte de la signature de la fonction lors de l’appel, le respect de type de return, ect.
* **Gestion de la table de symboles** La table doit avoir au minimum les champs suivants :
* Nom : l’identificateur qui indique le nom de la variable (ou constante), Tab, etc.
* Type : le type de la variable ou la constante
* Taille : la taille du tableau (la taille est égale à 1 pour les variables simples)
* **Génération des quadruplets.**
* **Traitement des erreurs :**

Il est demandé d’afficher les messages d’erreurs adéquats à chaque étape du processus de compilation, en précisant le numéro de ligne et la colonne de l’erreur.

**Bon courage.**