**Université des Sciences et de la Technologie Houari Boumediene**

**Faculté d’Electronique et d’Informatique**

**Département d’Informatique**

**Master SII, Année Universitaire : 2018/2019, Module Compilation**

1. **Introduction**

Le but de ce projet est de réaliser un mini-compilateur de langage décrit ci-dessous, en effectuant les différentes phases de la compilation : lexicale, syntaxique, sémantique, générations de code intermédiaire et génération de code objet.

1. **Description du Langage**

**II.1 La structure générale**

**#**PartieDeclarationBibliotheque

**Programme nomProgramme**

**{**

PartieDeclarationDeVariable

PartieInstructions

**}**

**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 |
| Calcul | Pour les instructions arithmétiques |
| BOUCLE | Pour les instructions de boucle |
| TAB | Lorsqu’on utilise des tableaux |

**La Partie déclaration de variables**

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

* Variable simple : Type  Liste de Variables;
* Tableau : Type  NomVariable [taille] ;
* Constante : Type CONST Liste de Constantes ;

**Liste de Variables**  est une suite d’identificateurs séparés par des ||

**taille** est un chiffre strictement positif.

**Un identificateur** est une suite alphanumérique qui commence par lettre **Majuscule** et qui ne dépasse pas 12 caractères.

**Type**est le nom du type de la variable qui peut être : Integer, Real.

La déclaration d’un ensemble de variables s’effectue en séparant les noms de variables par ‘,’.

Exemple :**Type ::** NomVariable1, NomVariable2;

|  |  |  |
| --- | --- | --- |
| Le type | Description | exemple |
| Ineteger | Un entier est une suite de chiffres. Signé ou non.Sa valeur est entre -32768 et 32767. | 5, -5 |
| Real | Une suite de chiffres contenant le point décimal. | 5.5, -5.5 |

**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 | **While (** cond)  **{**  Bloc Instructions  **}** | While(i<j)  {  A :=2 ;  A :=C+D ;  } |
| condition | **EXECUT** INSTRUCTIONS  **IF (**condition ) | EXECUT i :=5 ; IF (j==2) |

* **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
* **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 de code intermédiaire** sous forme de quadruplet. Et ceci pour toutes les instructions possible, en prenant en compte les boucles imbriqué.
* **Génération de code objet :** l’étudiant doit développer le code objet la structure assembleur 80x86.
* **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.**