TD/ TP : Analyse lexicale

Soit le programme suivant écrit dans un langage, auquel nous chercherons à créer un analyseur lexical.

Code 1

```
Début
      ma_Varaibale=12;
      b=2;
      si(var1!=b) alors{
            var1 = b;
      }
fin
```

A. L'analyse lexical

1- Quels sont les mots clé de ce langage ?
2- Définir les tokens correspondants aux mots clé de la question 1.
3- Donner les automates reconnaissants les tokens dans ce langage
4- Donner les expressions régulières qui permettent de définir les tokens de ce langage

B. Préparation de l'environnement de travail

1- Télécharger flex.exe sur le lien http://gnuwin32.sourceforge.net/packages/flex.htm
2- Installer flex dans le répertoire C:/
3- Ajouter le chemin C:\GnuWin32\bin dans la variable d'environnement Path

C. Création d'un analyseur lexical

1- Dans un fichier `unitesLexicales.h`, définir une valeur pour chaque tokens du langage
2- Déclarer dans le même fichier `unitesLexicales.h` deux variables externes « valEntier » et « valIdentif »
3- Créer un fichier .l et recopier le **Code 2.**
4- Créer un fichier .c et recopie le **Code 3.**
5- Lancer l'invite de commande et taper les commandes suivantes pour générer l'analyseur lexical

$ flex -omonCompilateur.c monFichier.l

$ gcc monCompilateur.c monFichier.Principal.c -o monCompilateur

6- Recopier le code du programme **Code 1** dans fichier .txt et taper la commande suivante pour tester votre analyseur

$ monCompilateur < programme.txt

7- Améliorer votre analyseur lexical pour reconnaitre tous les tokens du programme et supprimer les caractères de décoration.

Code 2

```
%{
#include <string.h>
#include "unitesLexicales.h"
%}
nbr [0-9]
entier {nbr}+
identif [a-zA-Z_][0-9a-zA-Z_]*
%%

debut       { ECHO; return DEBUT; }
fin         { ECHO; return FIN; }
{entier}    { ECHO; valEntier = atoi(yytext); return ENTIER; };
{identif}   {ECHO; strcpy(valIdentif, yytext); return IDENTIF; }
.           { ECHO; return yytext[0]; }
%%

int valEntier;
char valIdentif[256];

int yywrap(void) {
    return 1;
}
```

Code 2

```
/*Mon Fichier principal .c */

#include <stdio.h>

#include "unitesLexicales.h"


int main(void) {

    int unite;


    do {

        unite = yylex();

        printf(" (unite: %d", unite);

        if (unite == ENTIER)

            printf(" val Entier: %d", valEntier);

        else if (unite == IDENTIF)

            printf(" Nom Identif :  '%s'", valIdentif);

        printf(")\n");

    } while (unite != 0);

    return 0;

}
```