

Partie lexical :

En entrée :

La formule de l'utilisateur:

- la casse n'importe pas (majuscules ou minuscules sont autorisées)
- l'utilisation des opérateurs est restreinte à deux opérandes. exemple : $(5+(4+x))$ est une expression autorisée, mais pas $(5+4+x)$.

- fonctions:

- sinus (sin)
- sinus cardinal (sinc)
- valeur absolue (abs)
- racine carrée (sqrt)
- partie entière (ent)
- logarithme népérien (log)
- cosinus (cos)
- tangente (tan)
- exponentielle (exp)

- variable:

- x

- opérateurs:

- plus (+)
- moins (-)
- multiplication (*)
- division (/)
- puissance (^)

- réels:

- tout réel possible (partie entière.decimal)

- pi (renvoie réel comme type)

- autres caractères:

- parenthèses
- crochets

La structure jeton est définie avec un type et une valeur. Elle définit également une longueur, utile uniquement à ce programme car elle permet de savoir combien de caractères passer après la détection d'une fonction ou d'un réel.

Le programme prend la formule qui est une chaîne de caractères et analyse caractère par caractère.

S'il reconnaît directement un caractère (opérateur, variable ou autre) il renvoie la structure jeton avec le bon type et la bonne valeur :

- type possible : OPERATEUR, VARIABLE, PAR_OUV, PAR_FERM, BAR_OUV, BAR_FERM
- Valeur possible : PLUS, MOINS, FOIS, DIV, PUIS

S'il reconnaît un nombre ou une lettre, il va regarder les caractères suivants afin de voir s'il s'agit d'un réel ou d'une fonction. Puis il renvoie la structure jeton correspondante, son type et sa valeur :

- type possible : REEL, FONCTION
- valeur possible : valeur du réel, nom de la fonction (explicité précédemment)

S'il ne reconnaît pas le caractère, il renvoie la structure jeton avec "ERREUR" comme type et 1 comme valeur (correspondant à une erreur lexicale).

Si le programme détecte plusieurs points (.) dans un même réel (ex: 91.3.7) il renvoie la structure avec "ERREUR" comme type et 2 comme valeur (correspondant à une erreur syntaxique).

Lorsque le programme a parcouru tous les caractères, il renvoie en dernier la structure jeton avec "FIN" comme type.

En sortie:

Un tableau de jetons contenant chaque éléments séparément avec leur type et leur valeur

Exemples d'utilisation :

Entrée : "sin(3+x)"

Sortie :

```
[["FONCTION", "sin"],  
["PAR_OUV", " "],  
["REEL", "3"],  
["OPERATEUR", "PLUS"],  
["VARIABLE", "x"],  
["PAR_FERM", " "],  
["FIN", " "]]
```

Entrée : "2.9*tan(\$)"

Sortie :

```
[["REEL", "2.9"],  
["OPERATEUR", "FOIS"],  
["FONCTION", "tan"],  
["PAR_OUV", " "],  
["ERREUR", "1"],  
["PAR_FERM", " "],  
["FIN", " "]]
```