

TLANG: TP Flex & Bison

Option Info

8 décembre 2017, 10h15 – 12h15

Vous trouverez les fichiers `eval.l` et `eval.y` sur le serveur pédagogique. Ils correspondent à la définition, respectivement, d'un analyseur lexical et d'un analyseur syntaxique pour des expressions arithmétiques sur des entiers.

Engendrer le programme

Pour engendrer le programme, vous pourrez utiliser les commandes suivantes¹ :

```
bison eval.y, qui produit les fichiers eval.tab.c et eval.tab.h
```

```
flex eval.l, qui produit le fichier lex.yy.c
```

```
gcc -c eval.tab.c, qui produit le fichier eval.tab.o
```

```
gcc -c lex.yy.c, qui produit le fichier lex.yy.o
```

```
gcc -o eval eval.tab.o lex.yy.o, qui produit le fichier exécutable eval
```

Le programme ainsi obtenu peut être exécuté par la commande `./eval`.

Questions

Pour tous les opérateurs que vous ajouterez, vous veillerez à bien prendre en compte leur priorité et leur associativité, via les règles de la grammaire.

Question 0. Engendrez, exécutez et testez le programme fourni ;

Question 1. Ajoutez la multiplication et la division (entière) ;

Question 2. Ajoutez la puissance² ;

Question 3. Ajoutez la prise en compte des entiers négatifs et du moins unaire (p.ex. $-(2+3)$) ;

Question 4. Ajoutez la prise en compte des nombres réels³.

Question 5. (*Pour les plus rapides*) Ajoutez la possibilité de réutiliser la dernière valeur calculée avec le mot-clé `ans` ;

Question 6. (*Pour les plus rapides*) Ajoutez la possibilité de réutiliser les 100 dernières valeurs calculées avec la syntaxe `ans[n]` ;

1. Le cas échéant, on utilisera `g++` à la place de `gcc` pour compiler du C++.

2. Utilisez la fonction `pow` qui se trouve dans `math.h`, car `^` est un ou exclusif bit à bit en C. En cas de problème d'édition de liens avec la fonction `pow`, ajoutez `-lm` à la fin de la commande d'édition de liens : `gcc -o eval...-lm`

3. Voir en particulier la section 3.4.1 du manuel de GNU Bison