

# Rapport TP2

---

## *Choix d'implémentation\**

### **Fonction bool isSymbol(char c) :**

- considère tous les symboles qui peuvent être utilisés pour les opérations
- comparaison en ascii

### **Fonction Queue\* stringToTokenQueue(const char\* expression) :**

- L'énoncé nous dit d'utiliser `curpos` en tant que `const` hors pour pouvoir incrémenter la position `curpos` ne peut avoir `const`.
- Pour que la fonction prenne en compte les valeurs à plusieurs chiffres il faut implémenter la boucle `while`  
`( *curpos != '\0' && !isSymbol(*curpos) ) { i++; curpos++; }` sans quoi une valeur comme 34 renverra les tokens "34" et "4"
- vérifier à chaque incrémentation de `curpos` que c'est pas la fin de la chaîne pour ne pas avoir un segmentation fault

### **Fonction void printToken(FILE\* f, const void\* e) :**

- Fonction map classique (complexité  $O(n)$ )

### **Fonction Queue\* shuntingYard(Queue\* infix) :**

- *Problème rencontré :*
  - Lors de l'implémentation de la condition `while ( ( there is an operator at the top of the operator stack with greater precedence ) or ( the operator at the top of the operator stack has equal precedence and the operator is left associative ) ) and ( the operator at the top of the stack is not a left bracket )` pour que tous les asserts soient validés il a fallu pour la dernière condition ajouter une condition ternaire `( ? : )` avec uniquement un `false`. Etant donné le `makefile` qui transforme tous les warnings en erreur la seule solution était donc `tokenIsParenthesis(stackTop(operator)) ? tokenGetParenthesisSymbol((stackTop(operator))) != ')' : true`

### **Fonction Token \*evaluateOperator(Token \*arg1, Token \*op, Token \*arg2) :**

- le cas où l'opérateur n'est pas pris en compte l'erreur est gérée et le programme interrompu.
- Si un argument ou l'opérateur n'est pas du bon type alors l'erreur est gérée et le programme interrompu.

### **Fonction void computeExpressions(FILE \*fd) :**

- gestion de l'erreur du `getfile`

### **Fonction int main(int argc, char \*argv[]) :**

- gestion de l'erreur du fclose

#### ***Description du comportement sur les jeux de tests fournis***

- Tous les jeux de test ont été validés

#### ***Analyse personnelle du travail effectué***

- Il subsiste toujours une fuite de mémoire (80bits) que je n'arrive pas à régler. La fuite est liée à un token soit-disant non libéré alors qu'il est dans une file qui est libéré plus tard.