
Ce projet est à faire en binôme. Il sera à rendre le mercredi 8 mars 16h.

Transformation et analyse de code C

Ce projet consiste à transformer des sources écrites en langage C pour qu'un ancien compilateur puisse les utiliser. Pour cela, les fichiers C doivent être analysés et transformés pour qu'ils respectent la norme utilisée par cet ancien compilateur. Les problèmes et les défauts de norme des fichiers sources que vous aurez détectés pendant l'analyse devront être récapitulés dans un fichier texte.

1 Les fichiers sources

Pour ce projet, vous devrez analyser un code C dont des exemples vous sont fournis. Ils devront donc être reconnu par votre parseur. Voici, par ordre d'importance, la liste des éléments du langage C à reconnaître :

1. Les éléments de base : le **main**, les **for**, les **if**, les déclarations, les affectations, les types de base, les expressions arithmétiques et booléennes.
2. Les instructions complémentaires : les **while**, les **do..while**, les **case**, les incrémentations et décrémentations.
3. Les fonctions : les déclarations de fonctions, les appels de fonctions, les **return**.
4. Les expressions avancées : les pointeurs et les tableaux.
5. La documentation : les différents commentaires.
6. Les fichiers séparés : les **header**, les **define**.

Les trois premiers éléments sont obligatoires et le quatrième fortement conseillé.

Il est important de respecter cet ordre d'importance. Nous ne voulons pas voir une personne gérant les commentaires mais pas les fonctions.

2 La norme et les erreurs

Le fichier source produit devra respecter autant que possible la norme fournie. Lors de votre analyse de ce fichier, vous devrez récapituler les défauts de norme dans un fichier texte. Les principaux défauts de norme à détecter, par ordre d'importance, sont les suivants :

1. Trop de paramètres.
2. Retours de fonctions différents de **int** ou **void**.

3. Nombre de fonctions par fichiers.
4. Nombre de lignes par fonctions et par fichiers.
5. Problème d'espacement.

Les trois premiers éléments sont obligatoires et le quatrième fortement conseillé.

Afin de pouvoir utiliser cet ancien compilateur, il vous faudra supprimer les **for** et les remplacer par des **while**.

Vous pourrez détecter aussi des erreurs dans le code dont telles que :

1. Une instruction après un **return**,
2. Pas de test du retour d'un **malloc**.

Ces éléments sont fortement conseillés.

Vous produirez un fichier texte permettant de résumer l'exécution de votre parser. Celui-ci sera découpé en deux parties : un résumé et une partie détaillée. La partie résumé indiquera simplement le nombre d'erreur/défaut de norme de chaque type détecté alors que le détail permettra d'afficher les lignes où se trouvent ces erreurs.

3 Les nouvelles sources

Le programme doit produire de nouvelles sources qui respectent la norme. Ainsi l'analyse du code produit doit passer sans encombre dans votre programme. Il faudra donc modifier ou supprimer les parties du code source pour lesquelles des erreurs ont été détectées.

4 Rendu

Le projet est à rendre pour le mercredi 8 mars 16h sur moodle. Il est à faire en binôme. Il est important de respecter l'ordre d'importance dans le travail à faire. Nous corrigerons par ordre d'importance. C'est à dire que nous nous arrêterons au premier point non fait même si certains points moins importants ont été réalisés.

La note sera décomposé en deux parties : une partie code et une partie soutenance. Dans la partie code, nous vérifierons si le code est bien écrit. La partie soutenance comptera pour la plus grande partie de la note. La soutenance aura lieu l'un des jours suivants : les mercredi 9, jeudi 10 et vendredi 11 mars. Elle sera individuelle et consistera à faire une démo du programme. Il vous sera demandé que le programme affiche l'arbre produit lors de l'analyse. Veuillez donc à obtenir un arbre le plus propre possible et sans noeud inutile. On vous posera aussi quelques questions sur votre code pour vérifier vos connaissances et votre travail. Il est donc formellement interdit de copier sur internet et/ou vos camarades. L'objectif, on vous le rappelle, est de vous apprendre à écrire un parser et non de vous apprendre à copier (vous êtes déjà des experts).