INFO0947 : Compléments de Programmation
Filtrer un Tableau

B. Donnet, G. Brieven
Université de Liège

1 Problème

Soit T, un tableau à N valeurs entières ($N > 0$) non nécessairement uniques. On désire construire une fonction qui permet de filtrer T.

Par filtrer, on entend éliminer du tableau toutes les valeurs qui ne vérifient pas une certaine propriété, notée p, et ce tout en conservant l'ordre initial du tableau. La valeur retournée par la fonction est la *longueur utile* du tableau `T` après filtrage (c'est-à-dire le nombre d'éléments de `T` qui respectent la propriété p).

Dans ce travail, nous vous demandons de construire la fonction suivante :

```
1 int filtrer(int *T, int N);
```

où `T` est un tableau de taille `N`, avec `N` > 0. Cette fonction modifie `T` en fonction de la propriété p et retourne la longueur utile de `T` après filtrage.

Nous vous demandons d'implémenter la fonction `filtrer()`en respectant les contraintes suivantes :
— il est interdit de passer par un tableau intermédiaire ;
— vous ne pouvez utiliser aucune librairie autre que `assert.h` ;
— votre solution doit être construite autour de **boucle(s) de type `while`**.
Soit la situation S_1 suivante (avec, pour le tableau `T`, `N` $= 7$ pour des raisons de lisibilité du document) :

T :	-1	5	-3	10	12	-2	-4

FIGURE 1 – Exemple de tableau `T`.

Soit la propriété p suivante :

$$p(x) \equiv x \geq 0$$

Soit la situation S_2 suivante :

T :	5	10	12	0	0	0	0

FIGURE 2 – Exemple de tableau `T`.

Partant de S_1, la fonction `filtrer` devra aboutir à S_2 en appliquant p à chacun de ses éléments. La valeur retournée par `filtrer` (situation S_2) sera, ici, 3.

La fonction suivante (nous vous donnons l'interface, à vous de l'implémenter) vous permet de vérifier si un élément du tableau de `T` vérifie la propriété p [1] :

```
1 /*
2  * PréCondition : /
3  * PostCondition : test = p(x)
4  */
5 int test(int x);
```

1. Il s'agit donc de l'implémentation de la propriété p.

La fonction `test()` est déclarée dans un header appelé `propriete.h`. Pour des raisons de simplicité, le header `propriete.h` vous est fourni sur eCampus. Il est évident que, lors de nos tests, nous ferons varier l'implémentation de la propriété p (votre implémentation de `filtrer()`ne doit donc pas être tributaire d'une quelconque implémentation de p).

2 Travail à Réaliser

On demande, pour le module `filtrer()`, de résoudre le problème en suivant le cheminement suivant :

1. formaliser (= utiliser une notation mathématique concise et précise pour décrire) le problème. À cette fin, proposer de nouvelles notations et/ou de nouveaux concepts ;
2. spécifier le plus précisément et formellement possible le module demandé ;
3. donner une découpe en SP respectant les contraintes du problème donné et indiquer comment ils s'emboitent (si applicable) ;
4. spécifier complètement et formellement chaque nouveau SP ;
5. construire des morceaux de programme correspondant à chaque SP (en adoptant l'*approche constructive* vue au cours) ;
6. rédiger le module final ;
7. démontrer de manière rigoureuse et formelle [2] la complexité théorique, dans le pire des cas (i.e., en utilisant la notation de Landau), de votre solution (normalement $O(N)$).

Lors de la rédaction du code, veillez à respecter les principes de la **programmation défensive**.

3 Aspects Pratiques

Le travail à rendre se compose d'une archive `tar.gz` [3]. Votre archive portera le nom `filtrer-groupeXX.tar.gz`, où `XX` fait référence à l'identifiant de votre groupe [4].

Une fois décompressée, votre archive devra donner naissance à deux répertoires : `rapport/` (cfr. Sec. 3.1) et `code/` (Sec. 3.2).

Tout non-respect des consignes se verra sanctionné par 2 points en moins dans la note finale de votre projet [5].

3.1 Rapport

Votre rapport devra être rédigé via l'outil LaTeX(nous vous renvoyons à la formation donnée le 08/02/2022 pour la structuration de votre document et celle du 10/02/2022 pour la rédaction d'un document en LaTeX) en utilisant l'en-tête du template LaTeXdisponible sur eCampus (Sec. Projets).

Dans le template, vous trouverez une section « TITRE » délimitée par des commentaires. Il suffit de compléter les définitions des macros `\intitule`, `\GrNbr`, `\PrenomUN`, `\NomUN`, `\PrenomDEUX` et `\NomDEUX` par, respectivement, le titre du projet, votre numéro de groupe, vos prénoms et noms. Vous pouvez redéfinir la macro `\tablemat` pour insérer une table des matières.

Attention ! Ne modifiez pas les dix lignes appelées « *Zone protégée* ».

Une fois compilé, votre document LaTeXdevra produire un fichier PDF dont le nom sera `filtrer-XX.pdf`, où `XX` représente toujours votre numéro de groupe. Ce rapport ne pourra pas compter plus de **12 pages**. [6]

2. Toute justification impliquant un texte argumentatif en lieu et place d'équations ne sera pas lue. Expliquez tout de même votre raisonnement.
3. Nous vous renvoyons à la formation sur la ligne de commande, donnée au Q1, pour la compression des fichiers dans une archive `tar.gz`.
4. Pour rappel, il est interdit de changer de groupe entre les différents projets. Votre identifiant doit nécessairement être un nombre composé de deux chiffres (compris entre 01 et 50).
5. De plus, toute archive ne pouvant être décomposée (c'est-à-dire une archive corrompue) ou ne produisant pas les dossiers `rapport/` et `code/` lors de la décompression engendrera une note finale **nulle**.
6. Sans compter la page de garde, ni son verso.

Le dossier `rapport/` sera composé des éléments suivants :
— votre rapport au format `.tex` ainsi que toutes les sources utiles à la compilation ;
— votre rapport déjà compilé au format `.pdf`

Votre rapport devra contenir tous les éléments nécessaires à la compréhension de votre code source, présentés clairement et dans un ordre logique. Soit, entre autres :
— la formalisation du problème et la description des notations introduites ;
— la spécification formelle de votre module `filtrer()` ;
— votre découpe en Sp, la description de chaque Sp et comment ils s'emboitent (si applicable) ;
— la spécification formelle de chaque Sp (si applicable) ;
— si un problème nécessite l'introduction d'une boucle, il faudra fournir un dessin de la situation générale (Invariant Graphique) et en dériver un Invariant Formel ;
— les différentes étapes ({PréCondition} Init {Inv}, ...) de la construction de chaque Sp et la Fonction de Terminaison de chaque Sp (quand cela s'avère pertinent). Nous vous demandons donc d'appliquer **explicitement** la démarche constructive telle que vue au cours.

Soyez clairs et précis dans vos explications. N'hésitez pas à ajouter un schéma si vous le jugez nécessaire. Dans ce cas, expliquez-le : un schéma seul ne suffit pas !

Soignez votre orthographe : au delà de trois fautes, chaque faute vous fera perdre 0,25 points.

Nous vous renvoyons au séminaire donné par Patricia Tossings, le 08/02/2022, sur la rédaction de rapport pour vous guider au mieux dans vos efforts d'écriture (n'hésitez pas à consulter les slides de la formation).

Remarque : Pour inclure du code C dans un document LaTeX, voir la remarque publiée sur le forum eCampus (Projets – Informations Générales → Inclure du code dans le rapport + conseils latex).

3.2 Code Source C

Votre code source devra être rédigé en langage C. Votre code source devra être accompagné d'un `Makefile`[7] rédigé par vos soins, la commande `make` permettant la génération d'un fichier binaire exécutable appelé `filtrer`. Ce fichier exécutable sera situé dans le même répertoire que celui où se trouve le code source (c'est-à-dire le répertoire `code/`). Son exécution doit nous permettre de tester la validité de votre code. La fonction `main` de cet exécutable devra être implémentée dans un fichier appelé `main.c`.

Votre code source sera adéquatement découpé en headers et modules. Les différents sous-problèmes identifiés dans votre rapport devront apparaitre clairement dans votre code. La fonction `filtrer()` doit être déclarée dans un header nommé `filtrer.h` et doit pouvoir être compilée et utilisée sans la présence du fichier `main.c`.

Il est **interdit**[8] d'utiliser des fonctions ou procédures de librairies tierces (y compris la librairie standard du C), à l'exception de `assert.h`.

4 Agenda

4.1 Milestones

Pendant la durée du projet, nous vous proposons deux *milestones*. L'objectif de ces milestones est de rencontrer chaque groupe individuellement afin de discuter de l'avancement du projet et corriger le tir le cas échéant. Ces rencontres sont organisées durant resp. la 3e et la 2e semaines qui précèdent la remise du travail final. À l'issue de ces rencontres, aucune note ne sera affectée. L'objectif est donc de réaliser une évaluation formative de votre travail pour vous aider à réaliser le projet et à en tirer des enseignements.

Les deux milestones sont les suivants :

7. cfr le cours INFO0030, "Partie 2 – Chapitre 1 : Compilation".
8. Cette restriction ne s'applique pas au fichier `main.c` où est implémenté l'exécutable de test `filtrer`

— **semaine du 14/03/2022**. Le milestone portera sur les notations/formalisations du problème. Afin de permettre à l'équipe pédagogique de préparer au mieux le feedback, il vous est demandé de nous (i.e., Benoit Donnet et Géraldine Brieven) envoyer par email un petit document [9] (pdf, rédigé en LaTeX) décrivant les différentes notations que vous comptez introduire. La deadline pour l'envoi de ce document est le samedi 12/03/2022, 12h00. Vous devez spécifier, dans l'objet de votre email, la chaîne de caractères suivantes : `[INFO0947] Projet 1 - Milestone 1`

— **semaine du 21/03/2022**. Le milestone portera sur les Invariants de Boucle nécessaires à la réalisation du problème. Afin de permettre à l'équipe pédagogique de préparer au mieux le feedback,il vous est demandé de nous (i.e., Benoit Donnet et Géraldine Brieven) envoyer par email un petit document (pdf, rédigé en LaTeX) décrivant vos Invariants de Boucle (Invariant Graphique et Invariant Formel) que vous comptez introduire. La deadline pour l'envoi de ce document est le samedi 19/03/2022, 12h00. Vous devez spécifier, dans l'objet de votre email, la chaîne de caractères suivantes : `[INFO0947] Projet 1 - Milestone 2`

La participation à ces milestones est obligatoire. Aucun groupe ne peut s'y soustraire. Pour définir le moment auquel votre groupe viendra rencontrer l'équipe pédagogique, deux sujets de discussion ont été créés dans le forum du cours, sur la plateforme eCampus. Il vous suffit de lire et de suivre les instructions détaillées dans le premier message de chaque sujet.

Veuillez faire débuter les sujets des mails envoyés à l'équipe pédagogique par la chaine "**[INFO0947]**".

4.2 Deadline

Les archives `tar.gz` finales doivent être soumises sur la Plateforme de Soumission avant le **04/04/2022** à **18h00** (l'heure du serveur faisant foi). Toute soumission postérieure à cette date ne sera pas prise en compte. Comme l'heure de l'horloge du serveur de soumission et celle de votre ordinateur peuvent être légèrement différentes, il est **fortement déconseillé** de soumettre votre projet à 17h59 et 59 secondes : ne prenez pas de risque inutile.

9. Petit = 1 page.