Université de Strasbourg
Certification du logiciel Feuille de TP

Preuve de programmes avec Frama-C

I) Eléments de syntaxe

Syntaxe des annotations :

```
formula ::= expr
          | expr rel expr
          | formule ==> formule
          | formule <==> formule
          | formule && formule
          | formule || formule
          | \forall type ident ; formule
          | \exists type ident ; formule
rel ::=  == | != | < | <= | > | >=
```

Syntaxe de pré et post-conditions :

```
/*@ requires ...;
    assigns ...;
    ensures ...; */
```

Annotation des boucles :

```
/*@ loop invariant ...;
    loop assigns ...;
    loop variant ...; */
```

Le prédicat `\valid(x)` spécifie que `x` pointe vers une zone mémoire valide.
Le prédicat `\valid(&t[0..n-1])` spécifie que les zones mémoire `t[i]` pour `i` dans `[0... n-1]` sont valides.

Le prédicat `\separated(x, y, ...)` spécifie que les pointeurs `x` et `y` correspondent à des zones mémoires distinctes. Comme pour `valid` on peut également parler des cases d'un tableau.

On désigne le résultat de la fonction par `\result`.
On désigne l'ancienne valeur d'une variable `x` par `\old(x)`.

La commande `assigns` permet de spécifier quels sont les variables modifiées. Exemple `assings t[0..n-1]` permet de spécifier que le tableau t est modifié. La commande `assigns \nothing` permet de spécifier que la fonction ne réalise pas d'effet de bord.

Pour lancer Frama-C avec les plugins RTE et WP utiliser la commande :
`frama-c -rte -wp moncode.c`

Il n'est pas utile de créer une fonction `main`, on écrira uniquement les fonctions demandées sans utiliser de `#include` et dans des fichiers séparés.

II) Exercices

Programmer en C spécifier et prouver à l'aide de Frama-C les fonctions suivantes :

1. Calculer le minimum entre deux entiers.
2. Tester si tous les éléments d'un tableau sont nuls :

```
/*@
   requires ...;
   assigns ...;
   ensures ...;
*/
int all_zeros(int t[], int n) {
    /*@
       loop invariant ...;
       loop assigns ...;
       loop variant ...;
    */
     ...
}
```

3. Remplir un tableau avec une valeur donnée.

```
void array_fill(int t[], int n, int k) {
    ...
}
```

4. Copier le contenu d'un tableau dans un autre (les deux tableaux sont de même taille et sont pris en argument).

```
void array_copy(int s[], int t[], int size) {
    ...
}
```

   Indication : il faut penser à exclure les appels tels que array_copy(t, &t[1], n−1).

5. Tester si deux tableaux sont égaux.
6. Tester si un tableau est un palindrome.
7. Rechercher l'indice du minimum dans un tableau.
8. Rechercher l'indice d'un élément dans un tableau.
9. Échanger deux pointeurs sans utiliser de variable intermédiaire.