

Nom : .....

Prénom : .....

Note : ..... / 10

## □ Exercice 1

On considère la fonction `mystere` suivante en C :

```

1  int mystere(int n){
2      assert (n>1);
3      int candidat = 2;
4      while (n%candidat !=0) {
5          candidat = candidat+1;}
6      return candidat;}

```

1. Donner la valeur renvoyée par `mystere(42)`

L'appel `mystere(42)` renvoie 2.

2. Proposer une spécification aussi précise que possible pour cette fonction

Cette fonction renvoie le premier diviseur de l'entier  $n > 1$  donné en argument.

3. Prouver la terminaison de cette fonction.

La suite des valeurs prises par `n-candidat` est :

- positive car comme  $n > 1$  et `candidat` est initialisé à 2, avant la boucle  $n - \text{candidat} \geq 0$  et de plus, la condition d'entrée dans la boucle impose que `candidat` <  $n$  (puisque  $n \% \text{candidat} \neq 0$ )
- strictement décroissante car `candidat` augmente à chaque itération donc  $n - \text{candidat}$  diminue.

Donc  $n - \text{candidat}$  est un variant de boucle se qui garantit la terminaison.

4. Ecrire une version récursive de cette fonction en OCaml.

```

1  let premier_diviseur n =
2      let rec aux n c =
3          if n mod c = 0 then c else aux n (c+1) in
4      aux n 2;;

```

## □ Exercice 2 : Terminaison d'une fonction récursive

Montrer la terminaison de la fonction ci-dessous pour  $n \in \mathbb{N}^*$  :

```

1  let rec f n =
2      if n=1 then 0
3      else if n mod 2 = 0 then 1 + f (n/2)
4      else 1 + f (n+1);;

```

Si  $n \leq 2$ , alors la fonction termine en moins de 2 appels récursifs. Sinon,  $n > 2$  montrons alors que les valeurs prises par `n` lors des appels récursifs de *rang pair* est un variant. On note  $n'$  (resp.  $n''$ ) la valeur de `n` après le premier (resp. le second) appel récursif.

- Si `n` est pair et  $n > 2$ , alors en notant  $n = 2k$  ( $1 < k$ ). On a  $n' = k$  et  $n'' = k + 1$  et comme  $k + 1 < 2k$ ,  $n'' < n$ .
- Si `n` est impair alors en notant  $n = 2k + 1$  ( $k > 0$ ). On a  $n' = 2k + 2$  et  $n'' = 2k$  donc  $n'' < n$ .

Donc dans les deux cas, après deux appels récursifs la valeur de `n` diminue et donc la fonction termine.