Analyse d'algorithmes

Florian Bourse

0 Pour chacune des fonctions suivantes, déterminer si elles terminent ou non :



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

let rec u n =
  if n = 0 then 1
  else 1 + u ((n * 2) mod 15)

let rec v n =
  if n = 0 then 0
  else 2 * v ((n + 1) / 2)

let rec w n =
  n = 1 || (n mod 2 = 0 && w (n / 2))

let rec z n =
  if w n then 0
  else 1 + z (n + 1)
```

1 Décrire ce que calcule la fonction `mystere` et le démontrer.



```
let rec mystere x =
  if x = 0 then 0
  else 2 * x - 1 + mystere (n - 1)
```

2 Démontrer que les deux fonctions suivantes donnent toujours le même résultat :



```
let rec fact1 n =
  if n <= 1 then 1
  else n * fact1 (n-1)

let fact2 n =
  let rec fact2_aux acc i =
    if i >= n then acc
    else fact2_aux (i * acc) (i+1)
  in fact2_aux 1 1
```

3 Pour monter un escalier de n marches, on s'autorise à chaque pas à monter soit 1 soit 2 marches. Combien de manières différentes de monter l'escalier y-a-t-il ?