

TD2

mercredi 14 septembre 2022 14:15

Exercice 23 : Appartenir à une liste

Écrire une fonction qui détermine si un élément appartient ou non à une liste.

Exercice 24 : Min d'une liste

Écrire une fonction qui calcule le minimum d'une liste non vide, sinon lève une exception.

```
type List =
  | Nil
  | Cons of int * List
```

② Let rec `mem_List` = Fun List `el` → match List with
 | Nil → false
 | Cons(x, P) → if x = el then true else in

③ Let rec `min_List` = Fun List → match List with
 | Nil → max_int
 | Cons(x, P) → min x (min_List P)

Let rec `sum_List` = Fun List
 → match List with
 | Nil → 0
 | Cons(x, P) → x + sum_List P

type `repp` = Pas | Rep of int

Let rec `mem_List` = Fun List
 → match List with
 | Nil → Pas
 | Cons(x, P) → match mem_List P with
 | Pas → Rep(x)
 | Rep(y) → Rep(mem x y)

```
type abin =
  | F
  | Bin of abin * int * abin
```

let rec taille = fun a

→ match a with

| F → 0

| Bin(g, *, d) → taille g + taille d + 1

sed