

## Prédicats – Vision algorithmique

### 1) Listes et parcours générique de listes

(voir la session en console interactive)

### 2) Vérification d'un prédicat sur les éléments d'une liste

Voici deux fonctions Python :

```
def P(n):                def Q(n):
    return n % 2 == 0     return n ≤ 50
```

En utilisant ces deux fonctions, écrire des fonctions (et quelques tests pour ces fonctions) qui résolvent les problèmes suivants :

- indiquer si tous les éléments d'une liste L sont pairs
- indiquer s'il existe un élément de L qui est pair
- indiquer s'il existe un élément de L qui est impair
- indiquer si tous les éléments d'une liste L sont impairs

Y a-t-il des liens logiques entre les quatre propositions précédentes ?

Trouvez différentes façons de coder ces fonctions, retrouve-t-on les liens de la question précédente ?

### BONUS

Idem avec les problèmes suivants (pas en lien direct avec le cours) :

- compter les éléments pairs d'une liste L
- indiquer si deux éléments consécutifs de L sont pairs
- indiquer si trois éléments consécutifs de L sont pairs
- indiquer si  $m$  éléments consécutifs de L sont pairs ( $m$  est un paramètre de la fonction)
- inventer un problème !

## Prédicats – Vision algorithmique

### 1) Listes et parcours générique de listes

(voir la session en console interactive)

### 2) Vérification d'un prédicat sur les éléments d'une liste

Voici deux fonctions Python :

```
def P(n):                def Q(n):
    return n % 2 == 0     return n ≤ 50
```

En utilisant ces deux fonctions, écrire des fonctions (et quelques tests pour ces fonctions) qui résolvent les problèmes suivants :

- indiquer si tous les éléments d'une liste L sont pairs
- indiquer s'il existe un élément de L qui est pair
- indiquer s'il existe un élément de L qui est impair
- indiquer si tous les éléments d'une liste L sont impairs

Y a-t-il des liens logiques entre les quatre propositions précédentes ?

Trouvez différentes façons de coder ces fonctions, retrouve-t-on les liens de la question précédente ?

### BONUS

Idem avec les problèmes suivants (pas en lien direct avec le cours) :

- compter les éléments pairs d'une liste L
- indiquer si deux éléments consécutifs de L sont pairs
- indiquer si trois éléments consécutifs de L sont pairs
- indiquer si  $m$  éléments consécutifs de L sont pairs ( $m$  est un paramètre de la fonction)
- inventer un problème !