

Schémas de parcours et de recherche dans une séquence

ec : fonction → élément

// renvoie la valeur de l'élément courant

findeSéquence : fonction → booléen

// findeSéquence renvoie vrai si on est positionné sur la fin de la séquence

démarrer : action

// Effet : se positionne sur le premier élément de la séquence

// E.I. : indifférent

// E.F. : séquence non vide et ec est le premier élément de la séquence
séquence vide et on est positionné sur la fin de la séquence, ec n'est pas significatif

avancer : action

// Effet : se positionne sur l'élément suivant

// E.I. : ec est le k-ième élément de la séquence et on n'est pas sur la fin de séquence

// E.F. : - soit on n'est pas sur la fin de séquence et ec désigne le (k+1)ième élément
- soit on est sur la fin de la séquence et ec n'est pas significatif (on est sur la fin de séquence)

1 - Parcours avec traitement intégré de la séquence vide

```
démarrer
initialisationDuTraitement
tantque non findeSéquence
    traitementElémentCourant
avancer
ftantque
    terminaisonDuTraitement
```

2 - Parcours avec traitement de la séquence vide distingué

```
démarrer
selon findeSéquence
    findeSéquence :
        TraitementSéquenceVide
    non findeSéquence :
        initialisationDuTraitement
        répéter
            traitementElementCourant
        avancer
        jusqu'à findeSéquence
            terminaisonDuTraitement
fselon
```

3 - Recherche d'un élément vérifiant une propriété P :

```
Démarrer
tantque (non FindeSéquence) etpuis (non P(ec)) faire
    Avancer
ftantque
// FindeSequence ou alors P(ec)
selon FindeSéquence
    non FindeSéquence: // élément trouvé
                        // ec = premier élément de la séquence vérifiant P
    FindeSéquence: // élément non trouvé, on est sur la fin de séquence
fselon
```

4 - Recherche du premier élément vérifiant la propriété P, on sait qu'un tel élément existe :

```
Démarrer
tantque non P(ec) faire
    Avancer
ftantque
// ec = premier élément de la séquence vérifiant P
```