

Les listes chaînées

Solution à l'énoncé :

Soit les déclarations de types et la procédure *BIDULE* suivantes :

Type *ptElement* = *^Liste*

Type *Liste* = **STRUCTURE**

Info : Entier

Suivant : *ptElement*

Fin STRUCTURE

Procédure *BIDULE* (*E p* : *ptElement*, *S trouvé* : Booléen, *E x* : Entier)

Début

Tant que ($p^{\wedge}.info \neq x$) et ($p^{\wedge}.suivant \neq \text{nil}$) **faire**

$p \leftarrow p^{\wedge}.suivant$

Fin Tant que

Si ($p^{\wedge}.info = x$) **alors**

$\text{Trouvé} \leftarrow \text{vrai}$

Sinon

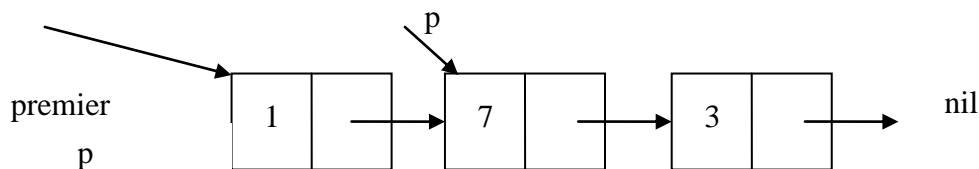
$\text{Trouvé} \leftarrow \text{faux}$

Finsi

Fin Procédure

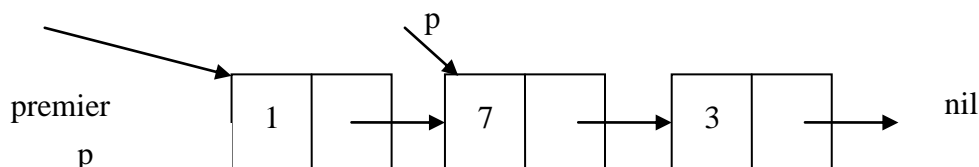
Solution à la question 1 :

1. L'appel de la procédure *BIDULE* (*q*, *trouve*, 7) qui se fait avec *q* (soit *premier*), *trouve* et *x* qui vaut 7
 - ⇒ La valeur ($p^{\wedge}.info = 1$) et elle est différente de $x = 7$ et aussi ($p^{\wedge}.suivant \neq \text{nil}$)
 - ⇒ *p* passe au suivant ($p \leftarrow p^{\wedge}.suivant$)

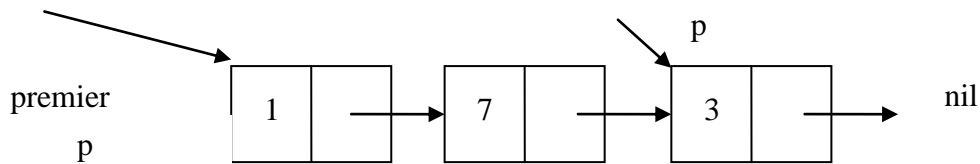


- ⇒ On recommence
- ⇒ La valeur ($p^{\wedge}.info = 7$) et elle est égale à x qui vaut 7. De plus ($p^{\wedge}.suivant \neq \text{nil}$)
- ⇒ On sort du tant que car une des deux conditions est vraie.
- ⇒ Comme ($p^{\wedge}.info =$ à la valeur de x), on affecte à *trouve* la valeur vrai.
- ⇒ On s'arrête.

2. L'appel de la procédure *BIDULE* (*q*, *trouve*, 12) se fait avec *q* (soit *premier*), *trouve* et *x* qui vaut 12
 - ⇒ La valeur ($p^{\wedge}.info = 1$) et elle est différente de $x = 12$ et aussi ($p^{\wedge}.suivant \neq \text{nil}$)
 - ⇒ *p* passe au suivant ($p \leftarrow p^{\wedge}.suivant$)



- ⇒ On recommence
- ⇒ La valeur ($p^{\wedge}.info = 7$) et elle est différente de $x = 12$ et aussi ($p^{\wedge}.suivant \neq nil$)
- ⇒ p passe au suivant ($p \leftarrow p^{\wedge}.suivant$).
- ⇒ On recommence



- ⇒ La valeur ($p^{\wedge}.info = 3$) et elle est différente de $x = 12$ mais ($p^{\wedge}.suivant = nil$)
- ⇒ On sort du tant que car une des deux conditions est vrai.
- ⇒ Comme $p^{\wedge}.info$ est différent de X , on va dans le alors
- ⇒ On affecte à $trouve$ la valeur faux.
- ⇒ On s'arrête.

Question 3 :

Que fait cette procédure ? Comment pourrait-on la renommer pour qu'elle soit plus explicite ?

Cette procédure permet de rechercher un élément dans une liste chaînée.

On pourrait la renommer **Procédure RECHERCHE** ((E) p : $ptelment$, (S) $trouvé$: $booléen$, (E) x : $entier$)

Question 4 :

En fait ici, lorsque l'on rentre dans le programme, on ne teste pas si $p \neq nil$. Or, si $p = nil$, le programme plante.

Pour cela il faudrait mettre une condition avant le tant que.

Procédure BIDULE (E p : $ptElement$, S $trouvé$: $Booléen$, E x : $Entier$)

Début

Si ($p \neq nil$) **alors**

Tant que ($p^{\wedge}.info \neq x$) et ($p^{\wedge}.suivant \neq null$) **faire**

$p \leftarrow p^{\wedge}.suivant$

Fin Tant que

Si ($p^{\wedge}.info = x$) **alors**

$Trouvé \leftarrow vrai$

Sinon

$Trouvé \leftarrow faux$

Finsi

FinSi

Fin Procédure