

Listes > Algorithmme avec allocation dynamique > Accès > Trace > Exemple 1

```
■ Accès(Liste_1, 6)
  Courant = L->Premier = 100
  Compteur = 0

  Courant = Courant = Valeur[Courant] =>Suisvant = 2520
  Compteur = 1

  Courant = Courant = Valeur[Courant] =>Suisvant = 300
  Compteur = 2

  Courant = Courant = Valeur[Courant] =>Suisvant = 140
  Compteur = 3

  Courant = Courant = Valeur[Courant] =>Suisvant = null
  Compteur = 4

Resultat = null
```

20/09/2006

1

Listes > Algorithmme avec allocation dynamique > Accès > Trace > Exemple 2

```
■ Accès(Liste_1, -1)
  Courant = null
  Resultat = null
```

20/09/2006

2

Listes > Algorithmme avec allocation dynamique > Insérer

```
■ Insérer : Liste ⊗ Entier ⊗ Element ⇒
  Liste
```

```
Fonction Insérer (L : Liste; Pos : Entier; Elt
: Element) : Liste;{
  Compteur : Entier;
  Courant : Pointeur[Place];
  Compteur = 1;
  Ins : Place;
```

20/09/2006

3

Listes > Algorithmme avec allocation dynamique > Insérer

```
Si (Pos = 0) Alors {
  Ins=>Element = &Elt;
  Ins=>Suisvant = L->Premier;
  L->Premier = &Ins;
}
Sinon{
  Courant = L->Premier;
  TantQue (Courant != null) Et (Compteur < Pos) Faire {
    Compteur = Compteur + 1 ;
    Courant = Valeur[Courant]=>Suisvant;
  }
  FinTantQue;

  Si (Courant != null) Et (Compteur == Pos) Alors {
    Ins=>Element = &Elt;
    Ins=>Suisvant = Valeur[Courant]=>Suisvant;
    Valeur[Courant]=>Suisvant = &Ins;
  }
  FinSi

}
FinSi
Retourne L
```

20/09/2006

4

Listes > Algorithme avec allocation dynamique > Succ > Trace

- Succ(&P_1)
Resultat = Valeur[P]⇒Suivant = 2520
- Succ(&P_4)
Resultat = Valeur[P]⇒Suivant = null

20/09/2006

5

Listes > Algorithme avec allocation contiguë > Créer_liste

```
□ Créer_liste : ⇒ Liste
Fonction Créer_liste : Liste {
  Resultat : Liste;
  Une_place : Place;
  Resultat⇒Premier = vide;
  Pour( I allant de 0 à Longueur-1 ) Faire {
    Une_place = Resultat⇒Places[I];
    Une_place⇒Libre = Vrai;
    Une_place⇒Suivant = Vide;
  }
  FinPour
  Retourne Resultat;
}
```

20/09/2006

6

Listes > Algorithme avec allocation contiguë > Accès > Trace > Exemple 1

□ Accès(Liste_2, 1)

Compteur = 0
Courant = L⇒Premier = 0

Courant = L⇒Places[Courant]⇒Suivant = 3
Compteur = 1

Resultat = L⇒Places[Courant] =
{3,192.168.0.40...,Faux,2}

20/09/2006

7

Listes > Algorithme avec allocation contiguë > Accès > Trace > Exemple 2

□ Accès(Liste_2, 5)

Compteur = 0
Courant = L⇒Premier = 0

Courant = L⇒Places[Courant]⇒Suivant = 3
Compteur = 1

Courant = L⇒Places[Courant]⇒Suivant = 2
Compteur = 2

Courant = L⇒Places[Courant]⇒Suivant = 1
Compteur = 3

Courant = L⇒Places[Courant]⇒Suivant = Vide
Compteur = 4

Resultat⇒Libre = Vrai;

20/09/2006

8

Listes > Algorithme avec allocation contiguë > Supprimer

- Supprimer : Liste \otimes Entier \Rightarrow Liste

Fonction supprimer(L : Liste; Pos : Entier) :
Liste;{

Compteur : Entier;

Prec : Entier;

Courant : Entier;

Compteur = 1;

20/09/2006

9

Listes > Algorithme avec allocation contiguë > Supprimer

```

Si (L=>Premier != Vide) Alors {
  Si (Pos == 0) Alors {
    L->Premier = L->Places[L->Premier]->Suivant;
  }
  Sinon {
    Prec = L->Premier;
    Courant = L->Places[L->Premier]->Suivant;
    TantQue (Courant != Vide) ET (Compteur < Pos) Faire {
      Prec = Courant;
      Courant = L->Places[Courant]->Suivant;
      Compteur = Compteur + 1;
    }
    FinTantQue
    Si (Courant != Vide) Et (Compteur == Pos) Alors {
      L->Places[Prec]->Suivant = L->Places[Courant]->Suivant;
      L->Places[Courant]->Libre=Vrai;
    }
  }
  FinSi
}
FinSi
}
FinSi
Retourne L;

```

20/09/2006

10

Listes > Algorithme avec allocation contiguë > Insérer > Trace > Exemple 1

- Insérer(Liste_2, 2, {192.168.1.50,...})
Compteur = 1
Compteur2 = 0
PlaceIns = vide

Compteur2 = 1

Compteur2 = 2

Compteur2 = 3

Compteur2 = 4

PlaceIns = 4
Compteur2 = 5

Courant = L->Premier = 0

Compteur = 2
Courant = Valeur[Courant]->Suivant = 3

L->Places[PlaceIns]->Element = EltIns = {192.168.1.50,...}
L->Places[PlaceIns]->Suivant = L->Places[Courant]->Suivant = 2
L->Places[PlaceIns]->Libre = Faux
L->Places[Courant]->Suivant = PlaceIns;

20/09/2006

11

Listes > Algorithme avec allocation contiguë > Insérer > Trace > Exemple 2

- Insérer(Liste_2, 5, {192.168.1.50,...})
Compteur = 1
Compteur2 = 0
PlaceIns = vide

Compteur2 = 1

Compteur2 = 2

Compteur2 = 3

Compteur2 = 4

PlaceIns = 4
Compteur2 = 5

Courant = L->Premier = 0

Compteur = 2
Courant = Valeur[Courant]->Suivant = 3

Compteur = 3
Courant = Valeur[Courant]->Suivant = 2

Compteur = 4
Courant = Valeur[Courant]->Suivant = 1

20/09/2006

Compteur = 5
Courant = Valeur[Courant]->Suivant = Vide

12