

**TD9 - Liste doublement chaînée (et circulaire)**

```

Type  liste2sens = Enregistrement
        tête, queue : ^cellule
        FinEnregistrement
cellule = Enregistrement
        val : élément
        succ, prec : ^cellule
        FinEnregistrement

Procédure Insérer (E/S l : liste2sens, E k : entier, E e : élément)
Var   r, p : ^cellule
        i : entier
Début
Si k=1
    Alors r ← allouer(cellule)
        r^.val ← e
        r^.succ ← l.tête
        r^.prec ← nil
        Si l.tête = nil
            Alors l.queue ← r
            Sinon l.tête^.prec ← r
        FinSi
    l.tête ← r
Sinon p ← l.tête
    i ← 1
    TantQue i<k-1 et p≠nil Faire
        p ← p^.succ
        i ← i+1
    FinTantQue
    Si p=nil
        Alors écrire ('Insertion impossible')
        Sinon r ← allouer(cellule)
            r^.val ← e
            r^.prec ← p
            r^.succ ← p^.succ
            p^.succ ← r
            Si r^.succ=nil
                Alors l.queue ← r
                Sinon r^.succ^.prec ← r
            FinSi
    FinSi
FinSi
Fin

Procédure Supprimer (E/S l : liste2sens, E k : entier)
Var   r, p : ^cellule
        i : entier
Début
Si l.tête≠nil
    Alors Si (k=1) Alors r ← l.tête
        l.tête ← r^.succ
        récupérer(r)
        Si l.tête = nil
            Alors l.queue ← nil
            Sinon l.tête^.prec ← nil
        FinSi
    Sinon p ← l.tête
    i ← 1
    TantQue i<k-1 et p^.succ≠nil Faire
        p ← p^.succ
        i ← i+1
    FinTantQue
    Si p^.succ=nil
        Alors écrire ('Suppression impossible')
        Sinon r ← p^.succ
            p^.succ ← r^.succ
            récupérer(r)
            Si p^.succ=nil
                Alors l.queue ← p
                Sinon p^.succ^.prec ← p
            FinSi
    FinSi
FinSi
Fin

```

### Liste doublement chaînée circulaire

```
Type liste2senscirc = ^cellule
cellule = Enregistrement
    val : élément
    succ, prec : ^cellule
FinEnregistrement
```

Procédure Insérer (E e : élément, k : entier ; E/S l : liste2senscirc)

```
Var r, p : ^cellule
i : entier
Début
r ← allouer(cellule)
r^.val ← e
Si l=nil
    Alors l ← r
        l^.succ ← l
        l^.prec ← l
    Sinon
        Si k=1
            Alors r^.suiv ← l
                r^.prec ← l^.prec
                l^.prec ← r
                r^.prec^.suiv ← r
                l ← r
            Sinon i ← 1
                p ← l
                TantQue i<k-1 Faire
                    p ← p^.succ
                    i ← i+1
                FinTantQue
                r^.prec ← p
                r^.succ ← p^.succ
                p^.succ ← r
                r^.succ^.prec ← r
        FinSi
    FinSi
Fin
```

Procédure Supprimer (E k : entier ; E/S l : liste2senscirc)

```
Var r, p : ^cellule
i : entier
Début
Si l≠nil
    Alors Si l^.succ=l
        Alors récupérer(l)
            l ← nil
        Sinon Si k=1
            Alors p ← l^.prec
            Sinon p ← l
            i ← 1
            TantQue i<k-1 Faire
                p ← p^.succ
                i ← i+1
            FinTantQue
            FinSi
            r ← p^.succ
            p^.succ ← r^.succ
            Si l=r
                Alors l ← r^.succ
            FinSi
            récupérer(r)
            p^.succ^.prec ← p
        FinSi
    FinSi
Fin
```