Le double hachage

```
Proc Rech(c :ent ; var trouv : bool, i :ent) ;
Var j,p : ent;
Debut
   Trouv = faux;  j=-1;      // j contiendra en cas de non trouv la première
                             // case effacée si rencontrée
   I=h(c) ; p=h'(c);         // P : le pas de décrémentation
         Tant que (non t(i).vide et non trouv)  faire
             Si t(i).val=c && non t(i).efface alors trouv= vrai
                  Sinon si j<0 && t(i).efface alors j=i ;
                           i=i-p ;  si i<0 alors i=i+N ;
             Fsi;
          Ftq;
          Si non trouv && j>=0 alors i=j ;
Fin.
```

La case vide reste une condition d'arrêt

```
Proc Supp(c) ;
Var i : ent ; trouv bool ;
Debut
   Rech(c,trouv,i);
   Si trouv alors
                t(i).efface=vrai ; nbIns—
   Fsi ;
Fin.
```

```
Proc Ins(c) ;
Var i :ent ; trouv : bool ;
Debut
   Rech(c,trouv,i) ;
   Si non trouv  && nbIns<n-1
        alors
            t(i)=(c,faux,faux) ;
            nbIns++
  Fsi
Fin.
```

Un pb à résoudre si la case i est la dernière case vide du tableau sachant que le nombre des cases vides décroit jusqu'à 1.

Que faire ?

La réponse :

Premièrement il faut commencer par détecter cette situation. Ensuite envisager une réorganisation du tableau qui remettra toutes les cases effacées à vide, On aurait pu penser à chercher une case effacée et la réutiliser. Dans ce cas on devrait déplacer la case vide, comme c'est impossible on s'arrête là.

La détection du pb :

Pour cela il faudrait connaître le nombre des cases effacées, car si il n'y a pas de cases effacées à quoi bon réorganiser. Il suffit de les comptabiliser au niveau de la suppression au lieu de décrémenter nbIns. On utilisera nbEff comme variable globale initialisée à zéro.

Modification de l'insertion et de la suppression :

Il faudrait tout simplement ne plus décrémenter le nbIns dans la suppression. Comme ça au moment de l'insertion si nbIns = n-1 il n'y aura pas d'insertion. Mais si l'insertion se fait dans une case effacée il faudra penser à ne pas incrémenter le nbIns car cette case est déjà comptabilisée.

L'Insertion devient :

```
Proc Ins(c) ;
Var i :ent ; trouv : bool ;
Debut
   Rech(c,trouv,i) ;
   Si non trouv
       alors
            Si  t(i).vide && nbIns<n-1 alors
               t(i)=(c,faux,faux) ;
               nbIns++
           sinon
               si t(i).efface alors   t(i)=(c,faux,faux)
           Fsi
  Fsi
Fin.
```