Exemple d'exercice de programmation

Cahier des charges :
Il est demandé de programmer le retrait d'un élément d'un tableau :
par exemple , programmer le retrait de l'élément d'indice i d'un tableau de n éléments.
(variantes possibles, retirer l'élément maximum, retirer l'élément minimum, retirer un élément tiré au hasard)

Remarques préalables :

Un exercice ou un projet de programmation comportera quatre parties (sauf s'il est immédiat à résoudre)

Partie I : Analyse
Partie II : Algorithme
Partie III : Code du programme
Parie IV : Jeux d'essais
Partie V : (optionnel) extensions et développements possibles.

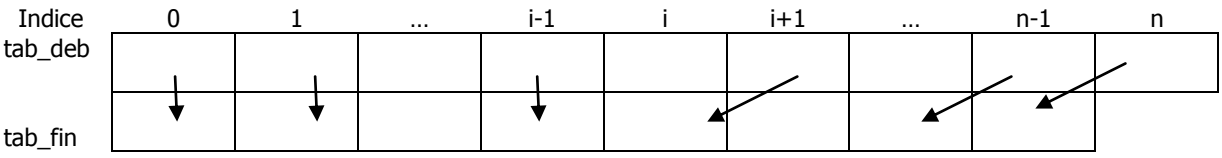
L'analyse est d'autant plus nécessaire que ce qui est demandé dans le cahier des charges est complexe.
L'algorithme doit être commenté.
Le programme doit être commenté.
Les jeux d'essais ont été choisis pour couvrir le plus possible de cas du problème

Pour la partie IV, un fichier (text, doc, pdf…) indique les variables testées en entrées et indique les résultats obtenus après exécutions…javascool ne permet pas aisément de réaliser la partie IV, on pourra donc la rédiger sur traitement de texte après exécutions des jeux d'essais (on peut utiliser des « copier-coller »)

I Analyse du problème :
Le tableau de départ étant de taille n, il faudra donc déclarer un tableau de taille n-1 comme résultat. Il faudra donc prévoir le cas ou le tableau initial est de taille 1, en retour il faudra renvoyé un tableau vide ou un message d'erreur…un tableau vide semble préférable car s'il doit être à nouveau rempli avec un nouvel élément pour des besoins de compléter l'application, il faudra rremodidier cette partie du travail.

Notons pour l'instant ***tab_deb*** le tableau initial et ***tab_deb[i]*** l'élément à retirer, ***tab_fin*** le tableau final
Dans le cas standard ou il y a plusieurs éléments dans le tableau initial, il semble apparaître trois cas :

- Cas 1 :Si i=0 (soit tab_deb[0] à retirer) il suffit de recopier, à partir de k=1, tab_deb[k]dans tab_fin_[k-1] , l'élément d'indice k de premier tableau devient l'élément k-1 du deuxième tableau
- Cas 2 :Si i=n (dernier élément) , il suffit de recopier tab_deb[k]dans tab_fin_[k] en s'arrêtant à n-1
- Cas 3 :Si 0 <i<n de 0 à i-1 les éléments recopiés ont le même indice, pour k de i+1 à n les éléments de tab_deb[k] sont recopiés dans tab_fin_[k-1]



Le choix de deux tableaux distincts plutôt qu'un seul qui se retrouverait modifié est plus simple en javascool car la taille d'un tableau est fixé lors de son allocation (commande new) et reste constante. On ne peut pas passer d'une taille n à une taille n-1.

II Algorithme :

```
Variables tab_deb ; tab_fin de type tableau d' entiers   //on prend un tableau de nombre pour l'exemple mais le principe est le même un tableau d'un autre type
          i,n sont entiers                                //indice de l'élément à retirer de tab_deb, n est la taille du tableau on suppose que n>=1.
          k est entier                                    // index des boucles
```

```
Debut
    Si n=1 alors tab_fin est le tableau vide
            sinon  pour k variant de 0 à i-1 faire
                    tab_fin[k] ←tab_deb[k]
                     fin pour
                    pour k variant de i+1 à n faire  // l'élément tab_deb[i] est évité
                    tab_fin[k] ←tab_deb[k-1]
                    fin pour
      finsi
fin
```

Remarque : avec l'expérience (ou simplement après avoir programmé) on s'aperçoit que le 3 ème cas inclus les deux autres car une boucle « pour i variant de 0 à -1 » est une boucle vide (ne faisant rien) …cela correspond au cas1 i=0 . Une boucle « pour i variant de n à n -1 » est une boucle vide (ne faisant rien) …cela correspond au cas2 i=n ;

III programme :

Il doit être donné plutôt (ou aussi) au format numérique . jvc pour javascool, .c pour le c ; .pas en pascal ;.py en python …

```
/*ce programme effectue le retrait de l'élément d'indice i
        du tableau tab_deb  et construit le tableau tab_fin
        qui comportera les éléments non supprimés.
        Pour tester la fonction demandée il faut créer
        un environnement comportant une fonction qui remplit
        un tableau initial et une fonction qui affiche
        le tableau final*/

void retrait_i(int tab_deb[],int tab_fin[],int i){
                /*retrait de l'élément d'indice i du tableau tab_deb ...
                 en fait construction d'un nouveau tableau tab_fin et
                 abandon du premier*/
        if (tab_deb.length>=1){
                for(int k=0;k<i;k++){
                        tab_fin[k]=tab_deb[k];
                }
                for(int k=(i+1);k<tab_deb.length;k++){
                        tab_fin[k-1]=tab_deb[k];
                }
        }
}

void init_tab(int tab[]){/*création d'un tableau avec des
                        valeurs aléatoires de 0 à 19,
                       on aurait pu aussi entrer des valeurs au clavier*/
        for (int i=0;i<tab.length;i++){
                tab[i]=random(0,20);
        }
}
void affich(int t[]){ //affichage à l'écran des éléments de t
        for (int i=0;i<t.length;i++){
                print(t[i]+"   ");
        }
        println("");
}

void main(){
        int n=10;int i=9;
        int tab_deb[]=new int [n];
        int tab_fin[]=new int [n-1];
        init_tab(tab_deb);
        affich(tab_deb);
        if (tab_deb.length>=1){
          tab_fin=new int [n-1];retrait_i(tab_deb,tab_fin,i);
                        /*l'appel effectif
```

```
                        de la fonction demandée est ici*/
            }
        if (tab_fin.length>=1){affich(tab_fin);}
}
```

IV jeux d'essais:

Essai 1 :
En changeant la ligne 32 avec int n=10;int i=5;
On obtient :

```
14  17  17  7  4  3  16  16  19  8
14  17  17  7  4  16  16  19  8
```

Essai 2 : retrait du premier élément (élément d'indice 0 !)
En changeant la ligne 32 avec int n=10;int i=0;
On obtient :

```
16  3  9  2  5  7  18  14  5  5
3  9  2  5  7  18  14  5  5
```

Essai 3 : retrait du dernier élément (élément d'indice n-1 !)
En changeant la ligne 32 avec int n=10;int i=9;
On obtient :

```
5  14  0  16  2  6  0  13  19  16
5  14  0  16  2  6  0  13  19
```

Essai 4 :le tableau de départ n'a qu'un élément
En changeant la ligne 32 avec int n=1;int i=0;
On obtient :

```
12
    /*pas d'élément affiché pour tab_fin
```