TD 3 – soutien – chaine

1. Javanais inverse

```
Fonction javanais-inverse (java : chaine) : chaine
Var i : entier
    franc : chaine
Debut
franc←java
i←2 {pour éviter un 'AV' en début de chaine}
TantQue i<lg(franc)-2 Faire
   Si franc(i)='A' et franc(i+1)='V' et franc[i-1] dans consonnes
                                    et franc[i+2] dans voyelles
      Alors franc←effacer(franc,i,2)
   FinSi
   i←i+1
FinTantQue
Retourner(franc)
Fin
```

Autre solution

```
Fonction javanais-inverse(java : phrase) : phrase
Var i,j : entier
    franc, ja : chaine
    sup : booléen
Début
franc←java
ja←java
i←-1
sup←vrai
Répète
   j←pos(ja,''av'')
   Si j≠0
      Alors
            Si (j=1 et ¬sup) ou (j≠1 et j≠lg(ja)-1 et (ja[j-1] dans
                                    consonnes et ja[j+2] dans voyelles))
                  Alors i←pos(franc,ja)+j-1
                        franc←supprimer(franc,i,2)
                        ja←copie(franc,i,lg(franc)-i+1)
                        sup←vrai
                  Sinon
                        ja←copie(franc,i+2,lg(franc)-i+1)
                        sup←faux
            FinSi
   FinSi
jusqu'à j=0
retourne(franc)
Fin
```

'jAVe trAVavAVaillAVe' => 'je travaille'

2. Position

On souhaite écrire la fonction `pos` définie sur le type `chaine`. `pos(c1, c2)` retourne la position d'une chaine `c2` dans une autre `c1` ; 0 si `c2` n'apparaît pas dans `c1.`

2.1. Expliquer en français le principe **itératif** de cette fonction.
On parcourt c1 pour y trouver le premier caractère de c2. Si on ne le trouve pas on retourne 0, sinon on regarde si les caractères suivants dans c1 correspondent à ceux de c2. Si oui, on retourne l'indice du premier caractère de c2 dans c1, sinon on recommence le principe avec l'occurrence suivante du premier caractère de c2 dans c1.

2.2. Ecrire en pseudo-langage la fonction **itérative** qui réalise cette opération.

```
Fonction pos (c1,c2 : chaine) : entier
Var trouve : booleen
    i,j,posdeb,res : entier
Début
res←0
j←1
trouve←faux
TantQue (j+lg(c2)-1<=lg(c1)) et not trouve Faire
    i←1
    TantQue (j+lg(c2)-1<=lg(c1)) et (c2[1]<>c1[j]) Faire
       j←j+1
    FinTantQue
    posdeb←j
    TantQue (i<=lg(c2)) and (j<=lg(c1)) et (c2[i]=c1[j]) Faire
       j←j+1
       i←i+1
    FinTantQue
    Si i>length(c2)
       Alors  trouve←vrai
              res←posdeb
       Sinon j←posdeb+1
    FinSi
FinTantque
Retourner(res)
Fin
```