```
> #Exo 1 : Opérateurs d'echelle
#Soient deux applications L ("lowering" ou "descendant") et R ("raising" ou "montant") sur l'ensemble des entiers naturels. On a, quel que soit l'entier naturel n, L(n+1)=n et L(0)=0, et R(n+1)=n+2, R(0)=0. On considère qu'une liste de la forme [X1,X2,...,Xn] où chaque Xi est soit L soit R représente la composition des applications X1(X2(...(Xn))...). Par exemple, [L,R,R,L,R,L] correspond à l'application n->L(R(R(L(R(L(n)))))).


#Q1 : Ecrire une procédure pour coder L.

#Q2 : Ecrire une procédure codant R.

#Q3 : Ecrire une procédure prenant en argument un entier n et une liste d'opérateur [X1,...,Xn] et qui renvoie le
#résultat du calcul de la liste appliquée à n. Note : la liste vide est considérée comme l'application identique.

>
```

8

```
> #Exo 2 : Opérateurs d'échelle généralisés. On considère maintenant deux opérateurs sur les entiers naturels qui sont
#paramétrés par un entier relatif d. On a L(d)(n+d)=n quel que soit n et L(n)=0 quel que soit n < d (remarque L(d)=0
#également), et R(d)(n)=n+d quel que soit n>0 et R(d)(0)=0. En particulier L(0) et R(0) sont l'application identique.
#Ecrire une procédure Echelle à trois arguments : une lettre X (soit L, soit R), un entier d et un entier n telle que
#Echelle(L,d,n) renvoie le résultat L(d)(n) et Echelle(R,d,n) renvoie le résultat R(d)(n).
```

5

```
> #Exo 3 : On considère des listes formées des lettres x ou y. On suppose qu'un y suivi immédiatement d'un x dans la liste #peut se réécrire comme la même liste où l'on a échangé les places de x et de y. Par exemple, [x,x,y,x,y,x] devient
#[x,x,x,x,y,y] lorsque l'on applique plusieurs fois cette réécriture. Ecrire un programme codant cela.



> #Exo 4 : On considère des listes formées des lettres x ou y. On suppose qu'un y suivi immédiatement d'un x dans la liste #peut se réécrire sous la forme 1+la même liste dans laquelle sont intervertis x et y. Par exemple, [x,x,y,x,y,x] devient
#1+[x,x,y,x,x,y], puis 2+[x,x,x,y,x,y], et enfin
```

```
3+[x,x,x,x,y,y]. Ecrire une procédure qui réalise ces
réécritures (tant qu'il y en a à faire!).
> #Exo 5 : Soit c un entier positif. On considère Ic l'ensemble
des entiers naturels inférieux strictement à c (en particulier,
I0 est l'ensemble vide). On définit une opération sur Ic union
{infinity} : x*y=x+y si x+y<c et x*y=infinity si x+y>=c, avec x
et y dans Ic, puis z*infinity=infinity=infinity*z quel que soit
z dans Ic union {infinity}.

Q1 : Ecrire une procédure "operation" à trois arguments x,y, et
c, où x,y sont dans Ic union {infinity} (on ne le vérifie pas)
et c est un entier, et qui renvoie le résulta x*y.
>
```

5

```
> #Q2 : Ecrire une procédure - dépendant de c - prouvant que
"operation" est associative.
>
```