CORRECTIONS DES EXERCICES DU CONCOURS D'ENTREE EPITA 2011

Pour toutes les fonctions demandées, plutôt que de fournir des solutions en C, Caml ou Pascal, nous avons préféré fournir une solution algorithmique en pseudo-langage facilement implémentable dans un de ces langages.

A) Arbre et généalogie…

1) C'est une numérotation hiérarchique *(le noeud **i** à pour fils gauche le noeud **2i** et pour fils droit le noeud **2i+1**)*.

2) Les numéros pairs sont des hommes et les numéros impairs des femmes.

3) L'ancêtre est de la $\boldsymbol{11^{ème}}$ génération.
   Oui c'est possible : (1, personne), (2, homme), (5, femme), (11, femme), (22, homme), (45, femme), (91, femme), (182, homme), (364, homme), (728, homme), (1457, femme)

4) Non, on ne peut pas. Il faut des arbres généraux parce qu'il peut y avoir plus de deux fils.

5) Degré de parenté entre deux frères: ***2***, entre deux cousins germains: ***4*** et entre une personne et le petit-fils de son oncle: ***5***

B) Factorielle calculable...

Spécifications :

La fonction ***plus_grand_pair(entier limite):entier*** retourne le plus grand entier pair dont la factorielle ne dépasse pas *limite*. Si *limite* ≤ *0* alors la fonction retourne *0*.

Principe :

A partir de *n=1*, on calcule toutes les valeurs successives de *n!* (en multipliant à chaque fois le résultat précédent par *n*) tant que l'on ne dépasse pas la valeur *limite*. Une fois la valeur limite dépassée, il suffit de retourner l'entier pair immédiatement avant *n*.

```
algorithme fonction plus_grand_pair : entier
parametres locaux
   entier limite
variables
   entier n, fact
debut
   n <- 1
   fact <- 1
   tant que fact <= limite faire
      n <- n+1
      fact <- fact * n
   fin tant que
   n <- n-1
   retourne (n - n mod 2)
fin algorithme fonction plus_grand_pair
```

C) Palindrome…

Spécifications :

La fonction ***est_palindrome (chaine phrase)*** retourne *vrai* si la chaîne *phrase* est un palindrome, *faux* sinon. La chaîne *phrase* ne contient que des minuscules non accentuées.

Principe :

Pour tester si une chaîne est un palindrome, il faut comparer les caractères symétriques. S'ils sont tous égaux, alors la chaîne est un palindrome. On se place donc sur le premier caractère (*pos=1*), et tant qu'on n'a pas dépassé la moitié de la chaîne et tant que le caractère en position courante (*pos*) est égal à son symétrique (en position *longueur(chaine)-pos+1*), on passe à la position suivante. Si on a parcouru la moitié de la chaîne alors c'est un palindrome.

```
algorithme fonction est_palindrome : booleen
parametres locaux
   chaine phrase
variables
   entier pos, long
debut
   long <- longueur(phrase)
   pos <- 1
   tant que (pos<long div 2) et (phrase[pos]=phrase[long-pos+1])
   faire
      pos <- pos+1
   fin tant que
   retourne (pos > longueur (phrase) div 2)
fin algorithme fonction est_palindrome
```

B) Liste vers 9…

Spécifications :

- La fonction ***liste_vers_9 (entier AB)*** affiche les éléments de la liste vers 9 générée et retourne la longueur (nombre d'éléments) de cette liste.

Principe :

On teste si AB est compris entre 10 et 99 et si les deux chiffres qui le composent sont différents (AB mod 11 = 0), Si ce n'est pas le cas, on affiche rien et on retourne *0*.
Dans le cas contraire et ce jusqu'à ce que AB=9, on affiche AB, puis on inverse les deux chiffres de AB que l'on soustrait à AB pour obtenir la nouvelle valeur de AB. On augmente alors la longueur de la liste de 1, longueur initialisée à 1.
Enfin on affiche la longueur de la liste.

```
algorithme fonction liste_vers_9 : entier
parametres locaux
   entier AB
variables
   entier long /* la longueur de la liste */
debut
   si (AB < 10) ou (AB > 99) ou (AB mod 11 = 0) alors
      retourne (0)
   sinon
      ecrire (AB, ", ")
      long <- 1
```

```
      faire
         AB <- AB - 10 * AB mod 10 + AB div 10
         si AB < 0 alors
             AB <- -AB
         fin si
         ecrire (AB, ", ")
         long <- long + 1
      tant que AB <> 9
      ecrire ("il y a ", l, " éléments.")
      retourne (long)
   fin si
fin algorithme fonction liste_vers_9
```