Deux exercices d'informatique : correction et un nouveau sujet

0.1 Le problème de Flavius Josèphe

Le programme :
$FJ := proc(n,p)$
$local\ k,T,res;$
$for\ k\ to\ n\ do\ T[k] := 1\ od;$
$res := NULL\ ; k := 0;$
$to\ n\ do$
$to\ p\ do\ k := suivant(k,T,n)\ od;$
$T[k] := 0;$
$res := res,k;$
$od;$
$res\ end;$
Pour la fonction *suivant* on appelle *ps* la place qui suit sur le cercle la place k : c'est $k+1$ si $k < n$ et 1 si $k = n$. Deux cas peuvent se présenter:
-si la place *ps* contient un élément ($T[ps] = 1$), c'est celui que l'on cherhe
-sinon il suffit de relancer la recherche à partir de *ps* par appel récursif $suivant(ps,T,n)$.

$suivant := proc(k,T,n)$
$local\ ps;$
$if\ k < n\ then\ ps := k+1\ else\ ps := 1\ fi;$
$if\ T[ps] = 1\ then\ ps\ else\ suivant(ps,T,n)\ fi$
$end;$

0.1.1

Bien que correct ce programme sature la pile de contextes lorsque n dépasse 1000 . Lorsque une fonction récursive est appelée, cet appel est stocké dans la pile, si l'exécution provoque un second appel à la fonction celui-ci est empilé par dessus le précédent et ainsi de suite. Supposons qu'il reste sur le cercle qu'un élément à la place n et que l'on exécute l'appel $suivant(1,T,n)$. Il produira l'appel $suivant(2,T,n)$, lequel produira l'appel $suivant(3,T,n)$, etc...jusqu'à $suivant(n-1,T,n)$. D'où la fonction non récursive :
$suivant(ps,T,n).\ suivant := proc(k,T,n)$
$local\ ps;$
$if\ k < n\ then\ ps := k+1\ else\ ps := 1\ fi;$
$while\ T[ps] = 0\ do$
$if\ ps < n\ then\ ps := ps+1\ else\ ps := 1\ fi\ od$
$ps\ end;$

0.2 Les palindromes

Pour retourner un entier $n = a_k a_{k-1} \cdot^{-} \cdot a_2 a_1 a_0$ il s'agit de calculer son retourné $\check{n} = a_0 a_1 a_2 \cdot^{-} \cdot a_{k-1} a_k$. Si l'on remarque que $\bar{n}$ est la valeur pour $x = 10$ du polynôme de coefficients les a_i, on voit que ce calcul est possible ou moyen de l'algorithme de Hörner, dès que les a_i sont donnés par indice croissant. Le chiffre des unités de n n'est autre que le reste de n dans la division par 10, alors que les autres chiffres sont le quotient. Faisons une même boucle ce qui donne :
$retourne := proc(n)$
$local\ n1,s;$
$n1 := n;\ s := 0;$

$while\ n1 <> 0\ do$
$s := 10 * s + irem(n1,10);$
$n1 := iquo(n1,10)$
$od;$
$s\ end;$

0.2.1

Faisons une boucle avec échappement :
$HautPal := proc(n : integer)$
$local\ j,n1,n2$
$n1 := n;$
$for\ j\ from\ 0\ to\ 100\ do$
$n2 := retourne(n1);$
$if\ n2 = n1\ then\ RETURN(j)\ else\ n1 := n1 + n2\ fi$
$od;$
$FAIL\ end;$

0.2.2

Il suffit alors :
$map(HautPal,[81..100]);$

0.2.3

En définissant de façon récursive l'indice palindromique I_p d'un entier par :
-si n est un palindrome, alors $I_p(n) = 0$
-sinon $I_p(n) = 1 + I_p(|n - \check{n}|)$
Ecrire la liste des palindromiques des entiers compris entre 1949 et 2009.

0.3

Ecrire un programme qui prend en arguments une matrice carrée A d'ordre 3 et un vecteur B de R^3, et qui reconnait, si possible, la transformation affine f de l'espace euclidien R^3 définie par $X \longmapsto AX + B$. On devra reconnaître :
-les projections : donner l'image et la direction, préciser si la projection est orthogonale.
-les symétries : donner les points fixes et la direction, préciser si la symétrie est orthogonale.
-les translations : donner le vecteur de translation.
-les rotations : donner un vecteur qui oriente l'axe et l'angle associé.
-les vissages : donner un vecteur qui oriente l'axe, l'angle associé et le vecteur de translation.