Projet algo :

Ce qu’il reste à faire

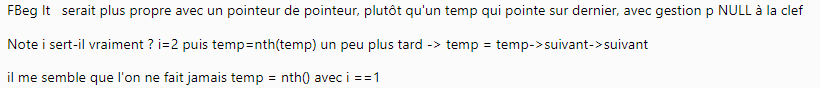
~~E~~   
  
~~Yn~~  
  
~~POW~~  
  
~~Ack~~  
  
~~Xn : RT : serait un peu plus propre avec un else /    m2 : maladresse : "if non test alors long sinon court" -> "if test alors court sinon long"~~

~~ZPUDT~~

~~Ai : Passage par adresse lourd et inutile~~

~~Test NULL dans surfonction lourd et inutile~~

~~PC~~  
  
~~NZAVPK rec, mettre else + m2 : maladresse : cf ci-dessus, j'arrête de le dire.~~  
~~NZAVPK it~~  
~~NZAVPK SF Test NULL dans surfonction lourd et inutile, j'arrête de le répéter~~  
~~NZAVPK SP~~   
  
~~NZAPRK : OIO : serait plus cohérent avec du out plutôt que du inout~~  
  
~~FBeg rec : lourdeur pour faire "return NULL"~~  
~~FBeg RT : n'est pas RT~~  
~~FBeg It iter : resultat à l'envers, donc copie de copie... affreux : pour faire une copie du cours de votre ami,  
vous faites une phocopie de son copie, puis comme c'est à l'envers, une phocopie de la photocopie ?~~

  
  
  
~~PBeg LD SDRE ; lourdeur save, depile, recursif, empile : si vous avez une photocopie à faire, vous en faites deux et vous jetez l'original ?~~

~~COMPTEURS POUR PERMUTATIONS~~

On peut simplifier la begaye rec ter

~~Ajouter rec ter a proc begaye~~