

Partie théorique : 50'

Question 1

6 (=3+3) points

- (1) Ecrire une *fonction récursive* **pgcd** qui calcule le plus grand commun diviseur de deux entiers naturels a et b . Indiquer les *limitations* éventuelles de l'algorithme.
- (2) Montrer, en précisant tous les appels récursifs, comment cette fonction calcule **pgcd(126,70)**.

Question 2

11 (=7+4) points

- (1) Ecrire une *fonction récursive* **puiss** qui calcule de façon aussi rapide que possible x^n , où x est un nombre de type *extended* et n un *entier naturel* de type *integer*. Indiquer les *limitations* éventuelles de l'algorithme.
- (2) Montrer, en précisant tous les appels récursifs, comment cette fonction calcule 3^7 .

Question 3

5 points

- a) Ecrire une fonction **horner** qui évalue un polynôme en un nombre x de type **extended** en utilisant l'algorithme de Horner. b) Expliquer comment est implémenté le polynôme dans Delphi dans cet algorithme.

Question 4

8 (=7+1) points

On donne la fonction booléenne **test** suivante :

```

function test(a,b:string):boolean;
begin
  if length(a)<>length(b)then result:=false
  else if a='' then result:=true
  else if pos(a[1],b)=0 then result:=false
    else begin
      delete(b,pos(a[1],b),1);
      delete(a,1,1);
      result:=test(a,b)
    end
end;

```

- (1) Calculer en détaillant :
 - a) **test('rose','tulipe')** b) **test('rose','ours')**
 - c) **test('roses','essor')**
- (2) Expliquer en une phrase ce que fait la fonction test en général.

Partie pratique : 70'

Un nombre palindromique (ou palindrome) est un entier naturel qui reste le même qu'on le lise de droite à gauche ou de gauche à droite. Par exemple : 47374, 101, 6 et 9449 sont des palindromes. Partant d'un entier naturel quelconque N , on obtient la **suite palindromique** associée à cet entier de la manière suivante :

- On inverse l'ordre des chiffres de N et on ajoute N à ce nombre inversé.
- On recommence l'opération avec le nouveau nombre jusqu'à ce qu'on obtienne un palindrome.

Exemples :

N	Suite palindromique
13	13, 44 (1 itération)
64	64, 110, 121 (2 itérations)
79	79, 176, 847, 1595, 7546, 14003, 44044 (6 itérations)
89	89, 187, 968, 1837, ... , 8813200023188 (24 itérations en tout !)
196	196, 887, 1675, ... (suite infinie ?)

On ne sait pas à ce jour si la suite palindromique de 196 se termine par un palindrome ou non, même après plus de 700 millions d'itérations (= répétitions) de renversement-addition ... Le but de ce problème est de développer une application Delphi avec une interface conviviale qui, à partir d'un entier naturel donné détermine sa suite palindromique. Le programme devra s'arrêter lorsque le nombre d'éléments de cette suite dépasse un seuil que l'utilisateur pourra fixer.

Programmation :

(1) Construire le formulaire ci-contre avec les éléments obligatoires suivants:

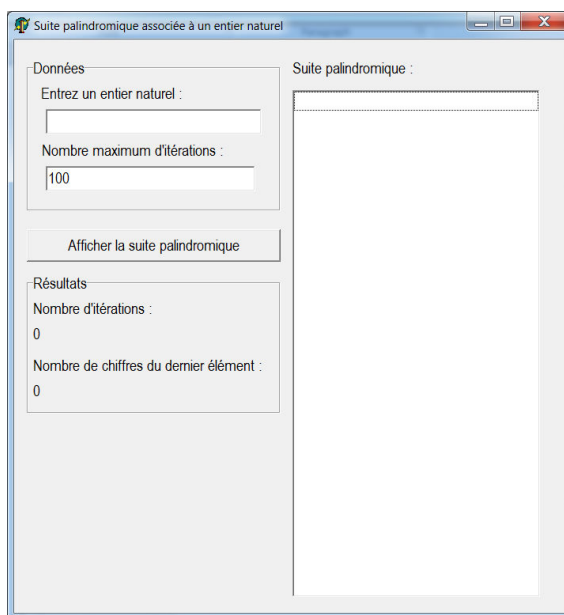
- **2 boîtes d'édition** : a) **edtN**, dans laquelle l'utilisateur pourra écrire un entier naturel et b) **edtMax** (valeur par défaut 100), qui contient le nombre maximum d'itérations.
- **2 libellés** : a) **lblIter** (valeur par défaut 0) qui contiendra le nombre d'itérations de l'opération de renversement-addition à faire jusqu'à obtenir un palindrome et b) **lblChiffres** (valeur par défaut 0), qui contiendra le nombre de chiffres du dernier élément de la suite.
- **1 liste** vide au démarrage, **lbSuite**, qui contiendra la suite palindromique.
- **1 bouton**, **btnCalcul**, avec la mention "Afficher la suite palindromique".

En outre le formulaire doit contenir les **libellés explicatifs** et les **boîtes de groupement** visibles ci-dessus.

8 points

(2) Ecrire une fonction **inverse**, qui retourne la chaîne de caractères inversée d'une chaîne de caractères donnée. Par exemple : **inverse('123') = '321'**.

4 points



- (3) Ecrire une fonction booléenne **palindrome**, qui vérifie si une chaîne de caractères donnée est un palidrome. Par exemple : **palindrome('12321')=True**.

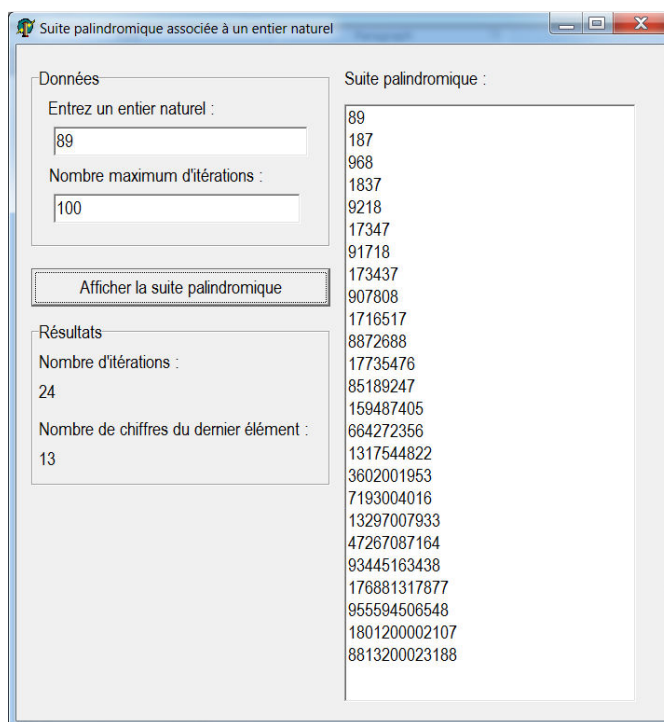
2 points

- (4) Ecrire une fonction **addinverse**, qui prend en entrée une chaîne de caractères représentant un nombre naturel, ajoute ce nombre au nombre inversé, et transforme à nouveau le résultat de l'addition en une chaîne de caractères. (Remarque : pour pouvoir traiter des entiers plus grands, on utilisera **strtoint64** au lieu de **strtoint** !)

4 points

- (5) Un clic sur le bouton "Afficher la suite palindromique" aura comme effet d'écrire les éléments de la suite palindromique dans la liste **lbsuite**. Si, après le nombre maximum d'itérations choisi (dans **edtMax**), l'élément correspondant n'est toujours pas un palindrome, le programme devra arrêter les calculs et afficher le message '**La suite palindromique ne s'est pas terminée !**' Dans tous les cas, le nombre d'itérations effectuées ainsi que le nombre de chiffres du dernier élément calculé devront être affichés dans les libellés prévus à cet effet (voir exemple d'exécution ci-dessous.)

12 points



G. Lorang