

## Partie théorique (30 points, 50 minutes, calculatrice autorisée)

## Question 1

17 (=6+3+4+4) points

- (1) Ecrire une *fonction booléenne* **premier** qui teste si un entier naturel donné est premier ou non.
- (2) Présenter dans un tableau d'exécution l'évolution des variables et du résultat de la fonction **premier** à l'appel de **premier(187)**.
- (3) Ecrire une *fonction récursive* **puissrapid** qui calcule de façon aussi rapide que possible  $x^n$ , où  $x$  et  $n$  sont des *entiers relatifs* de type integer. Préciser *tous les cas* éventuels dans lesquels cette fonction retourne un résultat faux !
- (4) Un entier  $n$  est un nombre premier de Mersenne si  $n$  et  $2^n - 1$  sont tous les deux premiers. (Par exemple 2, 3 et 5 sont des nombres premiers de Mersenne mais 11 n'est pas un nombre premier de Mersenne car  $2^{11} - 1 = 2'047$  est divisible par 23.) Ecrire une *procédure* **Mersenne** utilisant les fonctions des points précédents qui affiche à l'écran (dans une application en console) tous les nombres premiers de Mersenne compris entre deux entiers naturels  $a$  et  $b$  données en paramètre.

## Question 2

5 points

Ecrire une *fonction* **countOcc** qui compte *récursivement* le nombre le nombre d'occurrences d'une sous-chaine donnée dans une chaine de caractères donnée.

## Question 3

8 (=6+2) points

- (1) Ecrire a) une *fonction itérative* **pgcdI** et b) une *fonction récursive* **pgcdR** permettant de calculer le pgcd de deux *entiers naturels* par l'algorithme d'Euclide *par soustraction*.
- (2) Montrer comment la version récursive calcule le pgcd de 56 et de 16.

## Partie pratique (30 points, 70 minutes)

- Connectez-vous au serveur du LMR-L avec le login : **delphi.user** et le mot de passe : **Delphi1**.
- Créer sur votre bureau (desktop) un répertoire avec votre nom et prénom de la forme :  
**DEV1\_NOM\_PRENOM**
- Sauvegarder régulièrement votre application dans ce dossier.
- A la fin de l'épreuve, *imprimer le code-source* de votre application et *copier le dossier avec tous les fichiers dans mon IN*.

On demande de créer une application Delphi avec une interface graphique qui effectue certaines opérations sur des chaînes de caractères numériques.

- Reproduire fidèlement l'interface graphique de la *figure 1* en respectant les dénominations indiquées sur la figure. Ajoutez votre nom en tant que commentaire en haut du code source. (5 points).

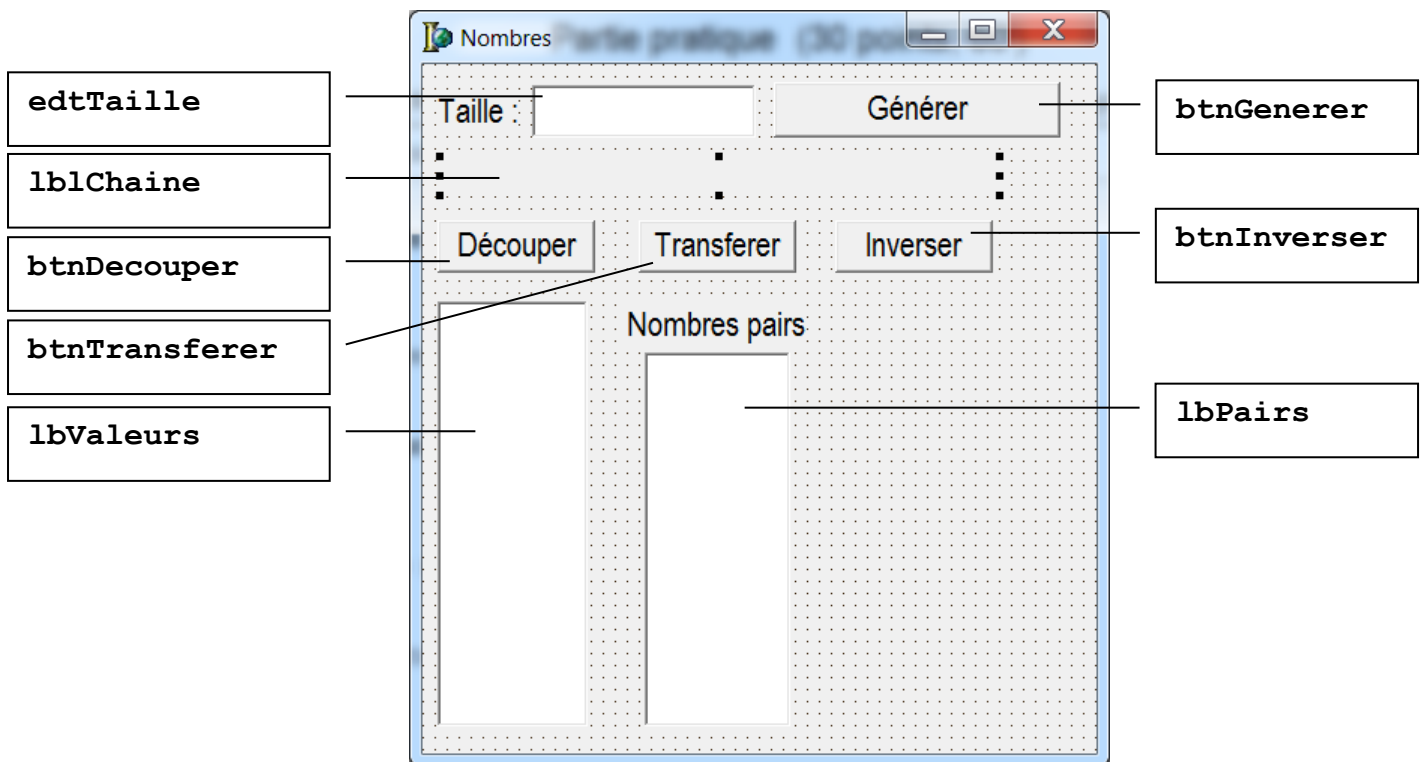
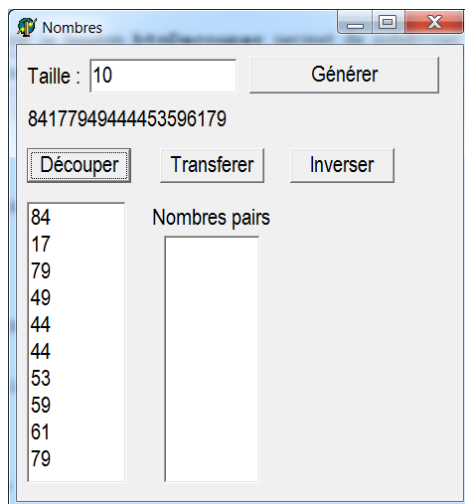


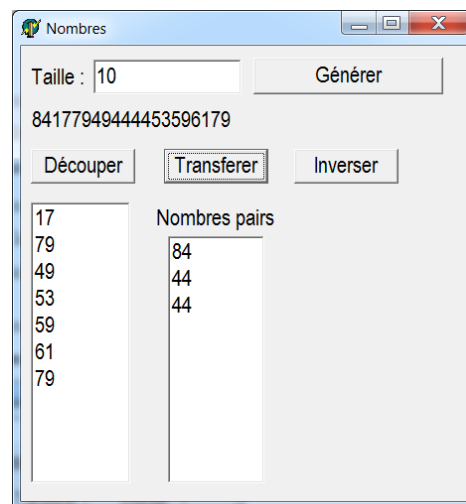
fig. 1

- Ecrire la fonction **alea**, qui prend en entrée un entier naturel  $n$  et qui retourne une chaîne de caractères contenant  $2 \cdot n$  chiffres aléatoires *non nuls*. Exemple : **alea(4)** pourrait retourner '28644915'. (4 points)
- Ecrire la fonction *réursive inverse*, qui prend en entrée une chaîne de caractères et inverse l'ordre des lettres de cette chaîne de caractères. Par exemple : **inverse('12345') = '54321'**. (4 points)

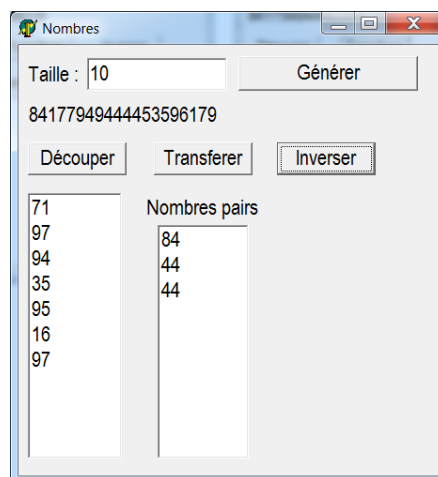
- (4) La boîte d'édition **edtTaille** permet de saisir un entier naturel  $n$ . Un clic sur le bouton **btnGenerer** produit, à l'aide de la fonction **alea**, une chaîne de caractères à  $2 \cdot n$  chiffres aléatoires non nuls et affiche cette chaîne dans l'étiquette (=label) **lblChaine**. De plus, les listes **lbValeurs** et **lbPairs** seront *vidées*. (3 points)
- (5) Un clic sur le bouton **btnDecouper** permet de subdiviser la chaîne contenue dans **lblChaine** en blocs de longueur 2. Ces blocs sont à ajouter à la liste **lbValeurs** (cf. *fig. 2*). (5 points)
- (6) Un clic sur le bouton **btnTransferer** permet de déplacer tous les nombres pairs de la liste **lbValeurs** vers la liste **lbPairs**, dans l'ordre. (Les nombres pairs doivent donc avoir disparu de la liste **lbValeurs** après ce clic) (cf. *fig. 3*). (6 points)
- (7) Un clic sur le bouton **btnInverser** permet d'inverser, à l'aide de la fonction **inverse**, tous les nombres de la liste **lbValeurs** (cf. *fig. 4*). (3 points)



*fig. 2*



*fig. 3*



*fig. 4*