

## Partie écrite (30 min)

## Question 1

14 points

- (1) Ecrire une *procédure* ou une *fonction* **Somme\_Poly** qui calcule la somme de deux polynômes donnés.
- (2) Ecrire une *procédure* ou une *fonction* **Produit\_Poly** qui calcule le produit de deux polynômes donnés.

## Question 2

8 (=2+2+2+2) points

Voici une fonction booléenne récursive **p** dont la définition inclut une fonction booléenne récursive *locale* **np**.

```
function p(n:integer):boolean;

    function np(n:integer):boolean;
    begin //début de la fonction np
        if n=0 then result:=false
        else result:=p(n-1)
        end; //fin de la fonction np

begin //début de la fonction p
    if n=0 then result:=true
    else result :=np(n-1)
end; //fin de la fonction p
```

- (1) Calculer **p**(6) en précisant toutes les étapes.
- (2) Calculer **p**(5) en précisant toutes les étapes.
- (3) Que calcule la fonction **p** en général ?
- (4) A-t-on le droit de définir la fonction **np** *avant* ou *après* la fonction **p** ?



**Programmation (38 pts) :**

- (1) Créer un *formulaire Delphi* (8 pts) avec les composants suivants :
- 1 boîte d'édition **edtN** dans lequel l'utilisateur entrera un entier  $N$  (valeur par défaut : 1000).
  - 3 listes :
    - a) Une liste **lbFrank** qui va contenir les éléments  $f(1), f(2), \dots, f(N)$  de la *suite de Frank* définie ci-dessus.
    - b) Une liste **lbDiff** qui va contenir les *différences successives*  $f(2) - f(1), f(3) - f(2), \dots, f(N) - f(N - 1)$ .
    - c) Une liste **lbPrem** qui va contenir les *éléments distincts de 1* de la liste **lbDiff**.
  - 3 boutons **btnFrank**, **btnDiff** et **btnPrem** sur lesquels l'utilisateur devra *cliquer dans cet ordre*<sup>2</sup> et dont l'effet est de remplir l'une après l'autre les trois listes ci-dessus. (3x3 pts)
  - 1 bouton **btnTest** dont l'effet est de tester si tous les éléments de la liste **lbPrem** sont des nombres premiers. Le résultat de ce test est à afficher dans un libellé **lblTest**. (4 pts)
  - 1 bouton **btnCount** dont l'effet est d'afficher *le nombre d'occurrences* d'un entier donné  $P$  dans la liste **lbPrem**. (5 pts). L'utilisateur précisera le nombre  $P$  dans une deuxième boîte d'édition **edtP**. Le résultat du comptage est à afficher dans un libellé **lblCount**.
  - Des libellés explicatifs pour chaque liste ou boîte d'édition.
- (2) Ecrire les algorithmes correspondants aux 5 boutons. L'application comportera obligatoirement les fonctions indépendantes suivantes :
- a) une *fonction booléenne* **premier** qui teste si un entier naturel donné est premier ou non (6 pts) ;
  - b) une *fonction* **pgcd** qui retourne le pgcd de deux entiers naturels (3 pts) ;
  - c) une *fonction* **f** qui retourne le  $n^{\text{e}}$  terme de la *suite de Frank* (3 pts).

G. Lorang

---

<sup>2</sup> On avertira l'utilisateur dans un libellé qu'il doit cliquer sur les boutons dans l'ordre, mais on n'aura pas besoin de le tester dans les algorithmes.