

# NF01 - Automne 2006

## Examen Final - 2 heures

**ATTENTION !**  
**Utilisez quatre copies séparées, une par problème**

### Problème n°1 (5 points) : Procédures et fonctions

On se propose de réaliser une version (très simplifiée) d'un distributeur de billets de train. Un client ayant choisi une gare de départ et une gare d'arrivée, ce distributeur doit calculer le prix du billet et simuler le paiement par pièces et le rendu de monnaie. Les gares à desservir sont (dans l'ordre) : Paris, Chantilly, Creil, Pont-Sainte-Maxence, et Compiègne. Elles sont éloignées de Paris de respectivement 41, 49, 60 et 76 km.

On connaît la loi qui gouverne le calcul des tarifs. Si  $d$  est la distance à parcourir, le prix en euros est de :

- $3 + 0,1 \times d$  si  $d < 70$  Km,
- sinon de :  $6 + 0,08 \times d$ .

- 1) Ecrire une fonction **PRIX** qui calcule le prix du billet (en l'arrondissant à l'euro inférieur) selon la gare de départ et la gare d'arrivée choisies par le client. (1.5 pt)
- 2) Ecrire une procédure **PAIEMENT** qui encaisse le prix du voyage. On attend de cette procédure qu'elle affiche, aussi longtemps que nécessaire, la somme qui reste à payer, et demande au client d'introduire une pièce autorisée (0,1€, 0,2€, 0,5€, 1€, 2€). Une fois le paiement effectué, elle doit afficher la monnaie à reverser au client et la renvoyer au programme principal. (2 pts)
- 3) Présentez le programme principal qui sollicite cette fonction et cette procédure. Vous avez toute liberté pour ajouter toute autre fonction ou procédure qui vous semblerait utile. Cette question sera appréciée selon la qualité de l'ensemble. (1.5 pt)

### Problème n°2 (5 points) : Défragmentation d'un fichier à accès séquentiel

Vous disposez d'un fichier composé d'enregistrements. Chaque enregistrement contient les informations suivantes : numéro d'ordre – type entier ; donnée – type chaîne de caractères. Ces enregistrements sont stockés dans un ordre quelconque dans le fichier du départ. L'objectif est de trier le fichier afin que les enregistrements soient classés à partir de leur numéro d'ordre.

**Question 1 :** Définissez le type de d'enregistrement et de fichier que vous devez manipuler.

**Question 2 :** Stocker dans un nouveau fichier l'ensemble des données, disponibles dans le fichier du départ, en respectant leurs numéros d'ordre (le premier à la première place, le deuxième à la deuxième place, ...).

**Question 3 :** Proposer un moyen simple (programme ou procédure ou fonction) qui permet d'accéder rapidement, dans le fichier obtenu en Q2, à une donnée identifiée par son numéro d'ordre.

### **Problème n°3 (5 points) : Récursivité**

**Question 1** (1 pt) : Soit la fonction récursive suivante

```
function f(n : integer):integer;
begin
  writeln(' n vaut ',n);
  if (n = 0) then
    f := 1
  else
    f := f(n+1);
  end;
```

La fonction f est appelée avec  $n \geq 0$ . Justifier la terminaison ou la non-terminaison de cette fonction.

**Question 2** (1 pt) : Soit la fonction récursive suivante

```
function g(n : integer):integer;
begin
  if (n <= 1) then
    g := 1
  else g := 1 + g(n-2);
  end;
```

La fonction g est appelée avec  $n \geq 0$ . Justifier la terminaison ou la non-terminaison de cette fonction.

**Question 3** (1 pt) : Soit la procédure récursive suivante

```
procedure h(t : array[1..10] of integer; n : integer);
begin
  if(n >= 0) then
    begin
      writeln(n, ' ', t[n]);
      h(t, n-1);
    end
  end;
```

La procédure h est appelée avec un tableau t d'au plus 10 entiers et  $0 \leq n \leq 10$ . Justifier la terminaison ou la non-terminaison de cette procédure.

**Question 4** (1pt) : Soit la définition de type suivante : `type tab = array[1..10] of integer ;`

Concevoir et écrire en pascal une fonction **somme\_tableau** de deux paramètres : un tableau de ce type **tab** et un entier **n** tel que  $0 < n \leq 10$ . Cette fonction effectuera récursivement la somme des éléments du tableau depuis l'indice **n** jusqu'à 0 et retournera cette somme.

### Question 5 (1pt) : soit la fonction suivante

```
function is_mot(chaine_en_cours : string):integer;
var ok_variable : integer;
begin
writeln(' chaine_en_cours ', chaine_en_cours);
if ((chaine_en_cours[1] = 'a') or (chaine_en_cours[1] = 'b')) then
begin
if (length(chaine_en_cours)<2)
then is_mot := 0
else
if ((chaine_en_cours[2] = '+') or (chaine_en_cours[2] = '-'))
then is_mot := 1
else
begin
ok_variable := is_mot(Copy(chaine_en_cours,2,length(chaine_en_cours)-2));
if ((ok_variable = 1)
and ((chaine_en_cours[length(chaine_en_cours)] = '+')
or (chaine_en_cours[length(chaine_en_cours)] = '-')))
then is_mot := 1
else
is_mot := 0;
end;
end
else is_mot := 0;
end;
```

La faire fonctionner avec **chaine\_en\_cours = 'abc+-'** puis avec chaine\_en\_cours = 'a++'. Que fait cette fonction récursive ?

## Problème n°4 (5 points) : Cours hors poly et TD

**Question 1** (2 pts) Ajout et suppression d'éléments dans une PILE et dans une FILE. Pensez à gérer les débordements par défaut (structure vide) et excès (structure pleine) dans la pile et la file.

0.5 pt : Ecrire en Pascal la procédure PUSHPILE permettant d'ajouter un élément dans une pile

0.5 pt : Ecrire en Pascal la procédure POPPILE utilisée pour retirer un élément d'une pile

0.5 pt : Ecrire en Pascal la procédure PUSHFILE qui ajoute un élément dans une file

0.5 pt : Ecrire en Pascal la procédure POPFILE servant à retirer un élément d'une file. Ces piles et files seront implantées par contiguïté, sous forme de tableaux.

**Question 2** (3 pts) Nous souhaitons gérer une liste de personnes (nom, prénom, âge, téléphone) implantée par chaînage (avec des pointeurs). La liste sera triée par ordre alphabétique sur le nom. Le début de la liste chaînée sera indiqué par le pointeur TETE.

1.5 pt : Ecrire en Pascal une procédure AJOUTE permettant d'insérer une nouvelle personne après la cellule pointée par le pointeur PTR, passé en paramètre.

1.5 pt : Ecrire ensuite la procédure EFFACE utilisée pour supprimer un élément placé après une cellule pointée par le pointeur PTR, passé en paramètre.