

# NF01 - Automne 2008

## Examen Final - 2 heures

(noté sur 21 au lieu de 20 => 1 pt de bonus !)

*Documents et calculatrices interdits*

### ATTENTION !

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

#### Problème n°1 (4 points) : Récursivité

On peut baser la multiplication de deux entiers naturels  $m$  et  $n$  sur l'addition et la division par 2. Le principe est alors le suivant :

- $m \times n$  est égal à 0 si  $n$  vaut 0
- $m \times n$  est égal à  $m + m \times (n - 1)$  si  $n$  est impair
- $m \times n$  est égal à  $(m + m) \times (n / 2)$  si  $n$  est pair

- 1) Ecrire une *fonction récursive* permettant de réaliser une telle multiplication de 2 nombres entiers.
- 2) Faire une simulation avec  $m = 3$ ,  $n = 5$ .

#### Problème n°2 (6 points) : Fichiers de texte

Après la rentrée, l'administration envoie un mail aux enseignants responsables des UV. Un fichier est attaché à ce mail contient, ligne à ligne, la liste des étudiants inscrits. Une ligne est une chaîne de caractère. Elle est composée d'un numéro d'ordre d'un nom, d'un prénom et d'un code pour le semestre. On supposera pour simplifier que les nom et prénom ne comportent pas de caractère blanc. L'objectif est de construire le fichier utilisable dans un tableur. Ce fichier, intitulé 'inscrits.txt', sera le résultat de l'exploitation par un programme pascal du fichier de données 'inscrits.txt'.

Un exemple de fichier 'inscrits.txt' :

```
1 BARBARIT OLIVIER TC03
2 BERTHET MEGANE TC01
3 BONNEMAISON DAVI TC03
4 CASQUIN ANTOINE TC01
```

Le nombre de caractères 'blanc' qui sépare les différentes parties est variable. L'enseignant copie ces lignes dans un fichier 'inscrits.txt'. Malheureusement, ce fichier ne peut pas être directement importé dans un tableur. Par contre le fichier suivant peut être directement importé dans un tableur :

```
;1;BARBARIT;OLIVIER;TC03
;2;BERTHET;MEGANE;TC01
;3;BONNEMAISON;DAVI;TC03
;4;CASQUIN;ANTOINE;TC01
```

En effet le caractère ';' peut être considéré comme un séparateur de cellule au moment de l'importation d'un fichier dans un tableur. Les caractères 'blanc' qui séparent les informations utiles ont été remplacé par un unique caractère ';'.

En résumé, à partir de la ligne :       « BARBARIT OLIVIER       TC03       »  
on cherche à obtenir la ligne :       « ;1;BARBARIT;OLIVIER;TC03 »

**Question 1** : écrire un programme pascal qui lit un fichier texte comme dans l'énoncé, dont le nom est demandé à l'utilisateur et qui affiche à l'écran son contenu.

**Question 2** : écrire la procédure `enleve_blancs_et_remplace_le_dernier_par_point_virgule(s1)` qui, si `s1` vaut ' 1 BARBARIT OLIVIER       TC03 ', retournera dans `s1` la chaîne ';1;BARBARIT;OLIVIER;TC03'

**Question 3** : écrire la procédure `init_compte_TC(T)` qui initialise à zéro toutes les cases d'un tableau `T` permettant de compter les TC par semestre. Définir le type du paramètre `T`.

**Question 4** : écrire la procédure `mettre_a_jour_compte_TC(s1,T)` où `s1` est une chaîne comme par exemple ';1;BARBARIT;OLIVIER;TC03' et `T` est un tableau qui permet de compter les étudiants par semestre et qui met à jour `T` en conséquence.

**Question 5** : à ce stade, vous considérez que les procédures ci-dessus existent et fonctionnent parfaitement. Construire les déclarations de variables et le programme principal qui lit un fichier 'inscrits.txt', crée un fichier 'inscrits.xtt' et affiche le nombre de TC01, TC02, etc. inscrits dans l'UV.

### Problème n°3 (5 points) : Ensembles / Le sac à dos

Vous partez en randonnée et votre sac à dos n'est toujours pas préparé. Vous avez développé un programme en Pascal qui vous permet de contrôler le remplissage et le contenu de votre sac à dos.

Votre programme présente les fonctionnalités suivantes :

1. Les objets que vous pouvez mettre dans votre sac à dos sont : `sac_couchage`, `chausseurs`, `gourde`, `chaussettes`, `linge`, `pain`. Donnez la définition d'un type ensemble « objets » composé de tous ces éléments.
2. Donner la procédure à un argument « `remplir_sac` ». Elle vous permet de déposer des objets (parmi ceux définis dans « objets »). L'utilisateur peut arrêter ou continuer le remplissage du sac après avoir déposé un objet. Cette procédure est appelée avec un sac vide et retourne un sac avec les objets déposés.
3. Donner la procédure « `affiche_sac` » qui affiche à l'écran les objets présents dans le sac à dos.
4. Donner la procédure « `vider_sac` » qui permet de sortir un ou plusieurs objets du sac en fonction de votre demande et leur présence dans le sac. Cette procédure est appelée avec un sac qui peut contenir des objets de l'ensemble « objets » et retourne le sac à dos sans les objets enlevés suivant le choix de l'utilisateur.
5. Donner le programme principal qui vous permet l'utilisation des différentes procédures et la manipulation de l'ensemble d'objets dans le sac à dos.

## Problème n°4 (6 points) : Procédures, fonctions, tableaux et enregistrements

Pendant votre stage TN09, on vous a donné pour mission le développement d'un système informatique pour la gestion des salariés d'une PME qui s'installe en Picardie. Les salariés sont caractérisés par leur nom, prénom, département (SRH, Etude, Fabrication) et leur fonction (ouvrier, secrétaire, ingénieur et directeur). Le cahier de charges est le suivant : inscrire tous les salariés actuels de l'entreprise dans le système informatique; rajouter des nouveaux salariés; exclure des salariés licenciés; modifier les caractéristiques d'un salarié.

Après une période de réflexion, vous proposez à votre tuteur d'écrire un programme Pascal avec:

- 1) Une procédure *saisie\_salarie* pour saisir les caractéristiques d'un salarié. Les données à saisir seront demandées à l'utilisateur.
- 2) Une fonction *enregistrement\_salarie* qui enregistre les salariés dans un tableau, passé par paramètre. La fonction demandera à l'utilisateur le nombre des salariés actuels et fera appel à la procédure *saisi\_salarie*.
- 3) Une procédure *modifier\_salarie* pour modifier des caractéristiques d'un salarié. Avant d'effectuer la modification, le programme demandera à l'utilisateur la confirmation du nom et prénom du salarié à modifier.
- 4) Une procédure *exclure\_rajouter\_salarie* pour exclure ou rajouter un salarié. Avant d'effectuer la modification, le programme demandera à l'utilisateur la confirmation du nom et prénom du salarié.
- 5) Une procédure *affiche\_salarie* qui affiche à l'écran tous les salariés d'un département donné.
- 6) Un programme principal qui propose à l'utilisateur un menu avec la possibilité de choisir une des actions précédentes;

Remarque importante : Les procédures des questions 2, 3 et 4 prendront le tableau et son nombre d'éléments en paramètres. La procédure de la question 5 ne prendra aucun paramètre.