



SERIE DE REVISION N°2
Algorithmique & Programmation



Lycée Bengurdenne

"Bac" Proposé par l'élève : Mahdhi Mabrouk "Info"

Exercice N°1:

Créer un fichier texte appelé exercice.txt et y stocker des valeurs (de type string) récupérées depuis le clavier jusqu'à la lecture de la chaîne 'FIN' (peut importe la casse, 'fin' ou 'Fin'). La chaîne de caractère, 'FIN' en l'occurrence ne sera pas stockée dans le fichier. Ecrire un programme qui récupère les 10 premiers mots du fichier, les stocke dans un tableau de 10 chaînes, puis affiche celui-ci.

Exercice N°2:

On propose ci-dessous l'algorithme d'une procédure de tri à bulles :

- 1) Procédure TRI_Bulles (d,f:entier ; Var T:tab)
 - 2) Pour i de d à f-1 faire
 - Pour j de d à f-i faire
 - Si T[j].nom < T[j+1].nom Alors Permute (T[j],T[j+1])
 - Fin si
 - Fin Pour
 - Fin Pour
- 2) Fin TRI_Bulles

Remarque :

Le module Permute (a,b) permute le contenu de deux entiers a et b.

Questions :

- a) Déterminer l'ordre du tri (croissant ou décroissant) accompli par la procédure TRI_BULLES.
- b) Dans le cas où le tableau T est déjà trié à la fin de la saisie, les parcours effectués par le module TRI_Bulles s'avèrent inutiles. En effet, aucune permutation n'aura lieu au sein de ce module dans ce cas. Modifier la procédure TRI_Bulles pour tenir compte de cette contrainte.
- c) Dédurre une solution récursive de cette méthode.

Exercice N°3:

Ecrire l'algorithme d'une fonction récursive intitulé « Somme » qui permet de calculer pour un ordre n donnée (n > 0), la somme approchée de la série définie par :

$$S = 1 + \frac{1}{5 \times 1!} + \frac{1}{25 \times 2!} + \dots + \frac{1}{5^n \times n!}$$

Exercice N°4:

Deux entiers N1 et N2 sont dits frères si chaque chiffre de N1 apparaît au moins une fois dans N2 et inversement.

Exemples :

- Si N1 = 1164 et N2 = 614 alors le programme affichera : N1 et N2 sont frères
- Si N1 = 405 et N2 = 554 alors le programme affichera : N1 et N2 ne sont pas frères

On se propose d'écrire un programme qui remplir et afficher un fichier intitulé **C:\freres.dat** comportant autant d'enregistrements (après chaque enregistrement nous testons la sortie par "Quitter (O/N) ?"). Chaque enregistrement est composé de trois champs successifs : deux champs pour deux nombres entiers aléatoire au maximum de l'entier et un champ pour sa vérification s'ils sont frères ou ne sont pas frères.

Exemples : Pour N1=1164 et N2=614 : Nature="sont frères"

Pour N1=405 et N2=554 : Nature="ne sont pas frères"

Questions :

- 1) Décomposer en modules le problème.
- 2) Analyser les différents modules ainsi le module principal.
- 3) Dédire les algorithmes correspondants.

Exercice N°5: (***) :**

L'utilisation des téléphones portables rendre facile la manipulation des personnes (nom et prénom) ainsi que leurs numéros. Chaque téléphone portable fournit un menu qui permet de gérer les enregistrements (personnes et numéros). Voici un exemple de menu

Répertoire	
Ajouter entrée	
Rechercher	
Copier	
Effacer	
Mes numéros	
Paramètres	
Afficher	
Select	Sortie

Chaque personne est représentée par nom (chaîne de 20 caractères), prénom (chaîne de 20 caractères) et numéro de téléphone (chaîne de 8 caractères). Ces informations sont enregistrées dans :

- un fichier à accès direct « TEL.DAT » si la mémoire utilisée est le téléphone
- un fichier à accès direct « SIM.DAT » si la mémoire utilisée est la carte SIM

L'action **paramètres** du menu offre la possibilité de choisir la mémoire. L'action **ajouter entrée** du menu permet l'ajout des personnes dans l'un des fichiers on arrête l'ajout des entrées si l'utilisateur n'a pas d'autres personnes à ajouter. L'action **rechercher** du menu permet de chercher un numéro de téléphone s'il existe :

- Dans le fichier« TEL.DAT » afficher « le numéro existe sur la mémoire de téléphone » ainsi que le nom et le prénom correspondants
- Dans le fichier« SIM.DAT » afficher « le numéro existe sur la carte SIM » ainsi que le nom et le prénom correspondants Si le numéro n'existe pas afficher « numéro inexistant » L'action **copier** du menu permet de copier des numéros du fichier« TEL.DAT » au fichier « SIM.DAT » et inversement. On peut copier un par un jusqu'à l'utilisateur demande d'arrêter la copie. L'action **effacer** du menu permet de supprimer des numéros du fichier« TEL.DAT » ou du fichier « SIM.DAT ». On peut supprimer un par un jusqu'à l'utilisateur demande d'arrêter la suppression ou de supprimer tous les numéros. L'action **mes numéros** du menu permet de copier des numéros du fichier « SIM.DAT » tous les numéros Tunisiens commençant par 20, 21, 22, appelé « TUNISIANA.TXT ». L'action **afficher** du menu permet d'afficher toutes les personnes dans un On peut quitter à chaque fois l'action sélectionné pour revenir au menu du sortir du menu pour à l'écran principal du téléphone.

Le programme permet de réaliser les actions de menu présenté ci-dessus.

Exercice N°6:

On dispose d'un fichier typé intitulé **annuaire.dat** qui contient les informations concernant un annuaire téléphonique. Chaque personne dans cet annuaire est caractérisée par un nom, prénom, date de naissance (jour, mois, année), numéro de CIN, adresse (numéro de rue, rue, code postal, ville) et numéro de téléphone.

Travail demandé :

Écrire un programme intitulé annuaire.pas doit nous permettre de :

- ☛ Saisir toutes les personnes dans le fichier **annuaire.dat**
- ☛ Afficher la liste des contacts dont le numéro de téléphone est palindrome et les mets dans un fichier texte intitulé palindrome.txt qui contient seulement le nom, prénom et le numéro de téléphone en question.

N.B : une chaîne se lit dans les deux sens de la même façon est dit palindrome.

☛ Permettre de rechercher un numéro de téléphone d'une personne donnée en utilisant un procédé récursif s'il existe sinon il affiche 'Ce contact n'existe pas dans la liste des contacts!'

Exercice N°7:(**) :**

Soit $A = (a_{ij})$ une Matrice d'ordre $n \times p$ et $B = (b_{kj})$ une matrice d'ordre $p \times q$. Le produit $A \times B = (c_{ij})$ d'ordre $n \times q$, est défini par $c_{ij} = \sum_{k=1}^p a_{ik}b_{kj}$.

On se propose d'écrire un programme permettant de faire les taches suivantes :

- ☛ Saisie contrôlée des tailles des matrices à fin de pouvoir calculer le produit.
- ☛ Remplir les matrices A et B par des entiers.
- ☛ Remplir la matrice C avec $C = A \times B$.
- ☛ Enregistrer la matrice résultante dans un fichier texte nommé **matrice.txt**.

Notez Simplement : L'enregistrement dans le fichier sera comme suivant :

Matrice.txt

```
La matrice résultante est :
-----
| 2 | 5 | 2 |
-----
| 1 | 6 | -1 |
-----
| 7 | -2 | 3 |
-----
```

Exercice N°8:(*) :**

1. Ecrire un module récursif permettant de remplir un tableau T de N chaîne.
2. Ecrire un module récursif permettant de retourner **ch** la plus longue chaîne.
3. Ecrire un module récursif permettant de trouver le nombre de répétition d'un caractère donné dans **ch**.
4. Ecrire un module récursif permettant d'afficher le nombre d'apparition de chaque caractère de **ch**.

Exercice N°9:(*) :**

Ecrire un programme permettant de trier et d'afficher les éléments entiers d'une matrice carrée M comportant N lignes et N colonnes ($3 < N < 11$) lus au hasard en utilisant la méthode de tri par insertion.

1. Décomposer ce problème en modules et analyser chacun des modules proposés.
2. Traduisez la solution en un programme pascal.

Exercice N°10:(***) (Bac Blanc Régional 2008 -Sfax):**

Pour déterminer le carré d'un entier naturel **n**, on utilise une méthode dont le principe est le suivant :

Le carré d'un entier naturel n est égal à la somme des n premiers entiers impairs.

Exemple : Pour $n=10 \rightarrow 10^2 = 1 + 3 + 5 + 7 + 9 + 11 + 13 + 15 + 17 + 19$

Travail demandé : Etablir l'analyse puis l'algorithme d'une fonction qui utilise la méthode décrite ci-dessus pour retourner le carré d'un entier naturel **n**.

- Proposez deux solutions :
 - a. Solution itérative.
 - b. Solution récursive.

Blague :

Comment un informaticien tente-t-il de réparer sa voiture lorsqu'elle a un problème? Il sort de la voiture, ferme toutes les fenêtres, retourne dans la voiture, et essaie de redémarrer.