

Exercice n°1 :

Une agence de voyage décide d'offrir à un client, parmi ces N clients fidèles ($2 \leq N \leq 10$), un voyage gratuit à l'étranger.

On détient pour chaque client fidèle les informations suivantes :

- un nom de 30 caractères au maximum
- une matricule chaîne composée que par des caractères alphabétiques et chiffres
- l'âge : entier non signé

Le choix du client gagnant se fait comme suit :

- on calcule pour chaque client un code appelé code de promotion
- Le client ayant le code de promotion le plus fort est le gagnant.

Le code de promotion d'un client est calculé à partir de la somme des chiffres du nombre extrait de sa matricule

- ❖ Le nombre extrait de la matricule est formé par la concaténation de tous les chiffres de la chaîne parcourue de droite à gauche. Si une Matricule ne contient pas de chiffres, le nombre extrait prend la valeur 0.
- ❖ La somme des chiffres du nombre extrait de la matricule est calculée comme suit :
 - On calcule la somme des chiffres du nombre extrait de la matricule
 - A cette somme on ajoute la somme des chiffres du nombre extrait de la matricule sauf le premier
 - Le calcul s'arrête dès qu'on termine tous les chiffres qui constituent le nombre extrait de la matricule

Exemple : pour un client qui dispose la matricule suivante : rt9x23gr1h

Le nombre extrait est 1329

Le code de promotion est calculé comme suit :

$$1+3+2+9 = 15$$

$$15+(3+2+9) = 29$$

$$29+(2+9) = 40$$

$$40+(9) = 49 \text{ (c'est le code de promotion)}$$

On se propose d'écrire un programme permettant de remplir un fichier **client.dat** par N ($5 < N < 100$) clients, puis de calculer et sauvegarder dans chaque ligne d'un fichier texte **voyage.txt** le nom et le code de promotion de chaque client.

Travail demandé :

1- Proposer une décomposition modulaire pour ce problème (algorithme)

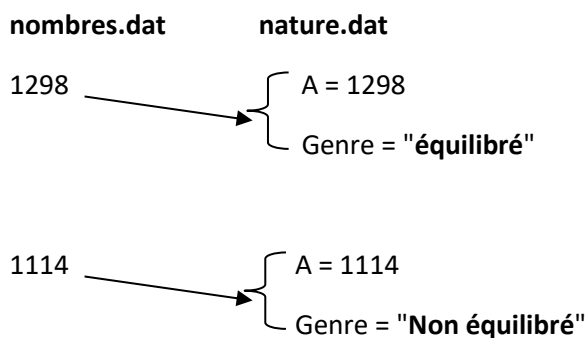
2- Ecrire les algorithmes des divers modules envisagés.

Exercice n°2 :

On dispose d'un fichier **nombres.dat** composé d'une suite de **N** entiers. On se propose de remplir à partir de ce fichier un autre fichier intitulé **nature.dat** contenant **N** enregistrements. Chaque enregistrement contient :

- **A** : l'entier lu à partir du fichier **nombres.dat**
- **Genre** : contient la chaîne "équilibré" ou "non équilibré".
 - « équilibré » si le nombre des chiffres pairs figurant dans **A** est égal au nombre des chiffres impairs. Exemples : 1298, 1889
 - « non équilibré » dans le cas contraire. Exemples : 1114, 4445

Exemples :



On se propose d'écrire un programme qui crée et remplit le fichier **nombres.dat** par **N** entiers ($3 < N < 31$).

Ensuite, il lit de ce fichier les nombres et remplit le fichier **nature.dat**.

Nb : les deux fichiers **nombres.dat** et **nature.dat** sont enregistrés sous la racine du lecteur **C**.

Exercice n°3 :

Soit **M** une matrice carrée d'ordre **N** ($2 < N < 15$).

On se propose d'élaborer une application permettant de générer puis afficher un fichier texte nommé « Nombres.txt » en appliquant la démarche suivante :

- Remplir une matrice **M** de la manière suivante :
 - Remplir la 1^{ère} ligne de la matrice **M** par des chiffres (entre 0 et 9) d'une façon aléatoire.
 - Pour les lignes suivantes, chaque case d'indice **L** et **C** est égale la somme des nombres pairs situés dans les cases situées dans la ligne qui la précède et les colonnes correspondantes variant de celle en cours **C** jusqu'à la dernière colonne (de **L-1** à **N**).

Exemple :

N=6 et M=

| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|-----|-----|----|----|---|---|
| 0 | 2 | 5 | 7 | 3 | 4 | 9 |
| 1 | 6 | 4 | 4 | 4 | 4 | 9 |
| 2 | 22 | 16 | 12 | 8 | 4 | 0 |
| 3 | 62 | 40 | 24 | 12 | 4 | 0 |
| 4 | 142 | 80 | 40 | 16 | 4 | 0 |
| 5 | 282 | 140 | 60 | 20 | 4 | 0 |

En effet :

- $M[1,0] = 2 + 4 = 6$
 - $M[4,1] = 40 + 24 + 12 + 4 + 0$
- Stocker dans chaque ligne du fichier texte la lettre "L" suivie par son numéro suivi par les valeurs des cases contenant des nombres pairs séparés par le caractère "#", puis la lettre "C" suivie par son numéro suivi par les valeurs des cases contenant des nombres impairs séparés par le caractère "#".

Travail demandé :

1. Proposer un algorithme modulaire du programme principal.
2. Proposer les algorithmes des modules nécessaires.