Série d'exercices N°2

Exercice 1

Soit le tableau de déclaration des nouveaux types ci-contre :

Nouveau Type
Tab= Tableau de 100 entiers

Dans le tableau suivant, transformer chacune des entêtes de ces procédures en une entête de fonction si cela est possible, sinon écrire « **Impossible** »et justifier cette réponse.

Entête d'une Procédure	Entête d'une Fonction et justification
procedure Proc1(Ch1 :Chaîne; @ Ch2:Chaîne)	…… …… …… ……
Procedure Proc2 (@ R : Tab ;T : Tab ; X : Entier)	…… …… …… ……
Procedure Proc3 (T : Tab ; X : Entier ; @ M : entier)	…… …… …… ……
Procedure Proc4(@ Test:Booléen)	…… …… …… ……

Exercice 2

On suppose qu'un programme principal contient trois sous programmes (une procédure **Proc1**, une fonction **Fonct** et une procédure **Proc2**).
Compléter le tableau suivant par un exemple d'appel de chacun des sous programmes dans le programme principal, en se basant sur les entêtes et sur la liste des variables globales disponible.

Entête du sous programme	Variables globales	Exemple d'appel du sous programme dans le programme principal
Procédure Proc1(@ m,n :entier ;x :réel)	A,b :entier X :réel Car :caractère Mot :chaine	
Fonction fonct(n :entier ;ch :chaine) :caractère		
Procédure Proc2(ch :chaine ; @c :caractère)		

Exercice 3 : nombre heureux

Ecrire l'algorithme d'un problème qui permet de saisir un entier **n > 0** puis de vérifier et d'afficher s'il est heureux ou non.
Un nombre heureux est un entier > 0, qui lorsqu'on additionne les carrés de chacun de ses chiffres, puis on additionne les carrés des chiffres de la somme obtenue et ainsi de suite, on obtient un entier à un seul chiffre et égal **à 1**

Exemple : pour l'entier 7 on a : $7^2 = 49$

$4^2 + 9^2 = 97$

$9^2 + 7^2 = 130$

$1^2 + 3^2 + 0^2 = 10$

$1^2 + 0^2 = 1$

On a obtenu un entier à un seul chiffre qui est égal à **1**, donc l'entier **7** est **heureux**

Exercice 4 :nombre pronique

Un nombre est dit pronique s'il est le produit de deux entiers naturels consécutifs

Exemples :

- ✔ 12 est un nombre pronique car 12=3*4
- ✔ 272 est un nombre pronique car 272=16*17
- ✔ 6006 =77*78

Un nombre est dit palindrome s'il se lit de droite à gauche comme de gauche à droite.

On désire écrire un programme permettant de lire un entier n strictement positif et déterminer et afficher si n est pronique et palindrome ou pronique seulement ou palindrome seulement ou n est un entier normal.

Exercice 5 :

Un nombre est dit super premier s'il est premier et si, en supprimant des chiffres à partir de sa droite, le nombre restant est aussi premier.

Exemple: Le nombre 59399 est super premier car les nombres 59399, 5939, 593, 59 et 5 sont tous premier.

- Ecrire l'algorithme d'un programme qui permet de :
  - ✔ saisir un entier n tel que 40000<n<100000
  - ✔ chercher et afficher tous les nombres super premiers inférieurs ou égaux à n.

Exercice 6 :

une chaine est dite existante dans un tableau de chaines si elle peut être formée à partir de la concaténation des $i^{ème}$ caractères des différents éléments de ce tableau.

Exemlpe: pour N=5 et le tableau T suivant:

"SALAH"	"AMIRA"	"BILEL"	"ANWAR"	"KARIM"
1	2	3	4	5

- ✔ pour **ch ="AMINA"** le programme affiche: **chaine existante dans T** car elle est le résultat de la concaténation des 2 ^{ème} caractères des différents éléments de T

✔ pour ch ="SALWA" le programme affiche: chaine inexistante dans T car les caractères de ch n'existe pas dans la même position dans les éléments de T

Travail demandé:

Ecrire un programme Python qui permet de saisir un entier **n(5<=N<=10)** et une chaine **ch** composée de lettres majuscules et de longueur **N**, puis de remplir un tableau **T** par **n** chaines composées de lettres majuscules et de même longueur que **ch** et de vérifier l'existence de **ch** dans **T** comme décrit ci_dessus.

Exercice 7:

Pour recharger un téléphone portable, un client doit composer ***XYZ*code de recharge#**

Un code de recharge est valide s'il vérifie les contraintes suivantes :

- Il est composé de **13 chiffres**.
- Le nombre formé par **les 3 premiers chiffres** est un **nombre premier**.
- Il contient **au moins une séquence de quatre chiffres** qui forment **les termes successifs d'une suite arithmétique** (la différence entre 2 chiffres successifs est la même).

Exemple de validité de code :

Soit l'essai de recharge suivante : *XYZ*1376357950241#

Le code **C**= **137**6357950241

- Le code **C** saisi contient 13 chiffres.
- Le nombre formé par les 3 premiers chiffres (**137**) est un nombre premier.
- Le code contient une séquence de 4 chiffres (3579) qui forment les termes successifs d'une suite arithmétique de raison 2.

D'où ce code **C** est **valide**.

Une fois le code **C** est valide, on passe à la vérification de son existence dans le tableau **Code** et la mise à jour de sa disponibilité dans le tableau **Etat** sachant que :

- **Code** est un **tableau de chaînes** contenant **n** codes de recharge **supposés distincts**(avec **10≤n≤100**).
- **Etat** est un **tableau de caractères** de même taille que le tableau **Code**. Chacune des cases peut contenir soit la lettre "**U**" si le code correspondant à la case du même indice du tableau **Code** est **U**tilisé, soit la lettre "**L**" si le code est encore **L**ibre.

On se propose d'écrire un programme permettant:

- de remplir le tableau **Code** par des **codes** valides selon les contraintes décrites ci-dessus.
- d'initialiser tous les éléments du tableau **Etat** à "**L**" c'est à dire que tous les codes du tableau sont au départ libres.
- de saisir un **code de recharge C** et de **vérifier sa validité** selon les contraintes décrites ci-dessus.
- de vérifier l'existence de C dans le tableau **Code** et de mettre à jour sa disponibilité dans le tableau **Etat** s'il est existant(**l'état du code passe de "L" à "U"**) ~~et~~
- d'afficher l'un des messages suivants :
  - **« Code invalide »** : Si le code n'est pas valide.
  - **« Code inexistant »** : Si le code est valide mais il n'existe pas dans le tableau **Code**.
  - **« Opération de recharge réussite »** : Si le code est **valide** et il est existant dans le tableau**Code**.

Travail demandé :

1) Ecrire l'algorithme du problème en le décomposant en modules.
2) Ecrire les algorithmes et les tableaux de déclaration relatifs aux modules envisagés.