MPSI/PCSI **TD d'informatique**

Pr. Youssef Ouassit

Algorithmique et programmation

« Les listes »

Exercice N° 1 :

Ecrire un programme qui lit 20 nombres entiers et qui affiche le résultat de la division de chaque nombre sur la moyenne de ces 20 nombres.

Exercice N° 2 :

Ecrire une fonction **« max_liste(L)** » qui détermine et retourne la plus grande valeur de la liste « L » passé en paramètre

Exercice N° 3 :

Ecrire une fonction « **conjuguer(verbe)** » qui affiche la conjugaison d'un verbe régulier en "er" au présent de l'indicatif. Contrôlez s'il s'agit bien d'un verbe en "er" avant de conjuguer, sinon on afficher message d'erreur.

Procédure à suivre :
Stocker les sujets dans une liste à initialiser à la déclaration.
Stocker les terminaisons dans une liste à initialiser à la déclaration.
Vérifier si le verbe se termine par « er », si non afficher une erreur.
Enlever « er » du verbe à conjuguer.
Afficher le verbe conjugué.

Exemple d'exécution pour le verbe « parler » :

```
Je parle
Tu parles
Il parle
Elle parle
Nous parlons
Vous parlez
ils parlent
Elles parlent
```

Exercice N° 4 :

Un polynôme sera représenté par deux listes, un pour les coefficients et l'autre pour les poids. Ecrire une fonction « **poly(C, D, x)** » qui calcule la valeur numérique du polynôme pour x. **C** est la liste des coefficients, **D** est la liste des poids

Exercice N° 5 : Opération sur les ensembles mathématiques

Préambule

Un ensemble mathématique est un groupement d'objets distincts, appelés éléments de cet ensemble. La théorie des ensembles est l'étude des propriétés et des opérations sur

des ensembles (appartenance, inclusion, réunion, . . .). Elle représente une branche essentielle des mathématiques.
Cet exercice s'intéresse aux algorithmes réalisant quelques traitements sur des ensembles mathématiques finis de nombres entiers.

Notation d'un ensemble fini d'éléments
Si E est un ensemble fini de N éléments (0 < N), e_0,e_1, . . . ,e_i ,e_{i-1}, . . . ,e_{N-1}, alors E sera noté ainsi E = {e_0,e_1, . . . ,e_i ,e_{i-1}, . . . ,e_{N-1}}.

Il s'agit de représenter les ensembles par des liste.

Question 1 Ecrire une fonction **appartient(L, a)** qui prend en argument une liste **L** d'entiers et un entier **a** et renvoie **True** si **L** contient **a** et **False** si non.

Question 2 Ecrire une fonction **union(L1, L2)** qui prend en argument deux listes **L1 et L2** d'entiers et renvoie l'union de L1 et L2.

Question 3 Ecrire une fonction **intersection(L1, L2)** qui prend en argument deux listes **L1 et L2** d'entiers et renvoie l'intersection de L1 et L2.

Exercice N° 6 :

Ecrire une fonction « **fusion(A , B)** » qui retourne le résultat de fusion des deux listes A et B supposés triés par ordre croissant. La liste retournée doit être trié par ordre croissant.
Exemple : Si A=[2, 5, 8, 9] et B=[1, 7, 10] la liste retourné sera [1, 2, 5, 7, 8, 9, 10]

Exercice N° 7 :

Ecrire une fonction **« trie (L)** » qui True si les éléments de la liste T sont trié dans ordre croissant ou False si non.

Exercice N° 8 :

Le crible d'Eratosthène est un procède qui permet de trouver tous les nombres premiers inferieurs à un certain entier naturel donné N.
L'algorithme procède par élimination : il s'agit de supprimer d'une table des entiers de 2 à N tous les multiples d'un entier. En supprimant tous les multiples, à la fin il ne restera que les entiers qui ne sont multiples d'aucun entier, et qui sont donc les nombres premiers. On commence par rayer les multiples de 2, puis à chaque fois on raye les multiples du plus petit entier restant. On peut s'arrêter lorsque le carré du plus petit entier restant est supérieur au plus grand entier restant, car dans ce cas, tous les non-premiers ont déjà été rayes précédemment. A la fin du processus, tous les entiers qui n'ont pas été rayés sont les nombres premiers inferieurs à N.
Ecrire la fonction erastothene(N) qui retourne la liste des nombre premiers inferieurs à N en utilisant la méthode du crible d'Eratosthène.