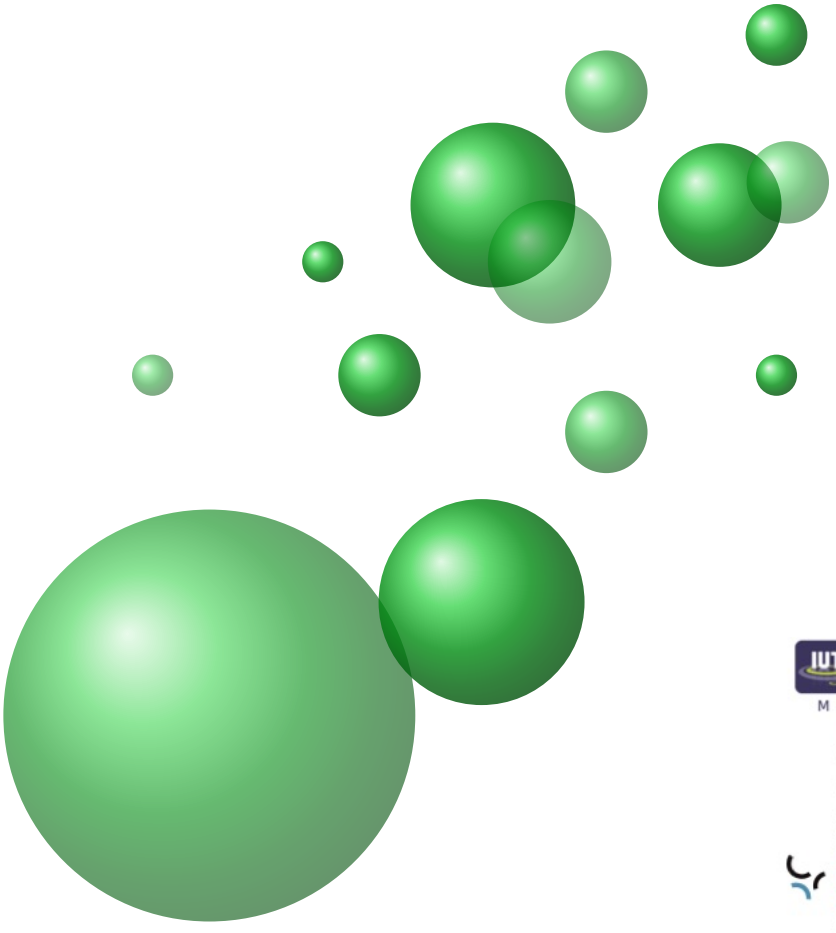
DUT INFORMATIQUE
IUT de Valenciennes
Antenne de Maubeuge

Ressource R1.01
Initiation au développement

Fiches de Travaux Dirigés

Philippe Polet
philippe.polet@uphf.fr

Année 2019– 2020



Semestre 1

Avant-propos

Ce recueil compile les sujets de TD de la ressource R1.01 Il se peut qu'en séance de TD tous les exercices ne soient pas traités. Il est fortement conseillé aux étudiants de les faire en autonomie. Bien que la majeur partie des exercices traités en séance soit corrigée, chaque étudiant doit être actif et doit essayer de les résoudre. Seule une démarche active en TD permet à l'étudiant de progresser.

Table des matières

Fiche de TD 1

Cette fiche de TD a pour but d'illustrer les éléments vus lors du premier cours.

Exercice 1

On considère le processus suivant : on choisit un nombre, on calcule son carré, on multiplie le résultat par 10 et on y ajoute 25.

Question A

Mathieu a choisi 2 comme nombre de départ et a obtenu 65. Vérifier par le calcul que son résultat est exact.

Question B

Rosalie affirme que si le nombre choisi au départ est un entier pair alors le résultat est pair. A-t-elle raison ? Justifiez.

Question C

Jean affirme que le résultat obtenu est toujours positif quelque soit le nombre choisi au départ. A-t-il raison ? Justifiez.

Question D

On suppose avoir à notre disposition la fonction *carre* telle que donnée en cours. Donner les algorithme du processus décrit sous la forme d'une fonction *calculer* qui prend en paramètre d'entrée une valeur réelle et qui retourne une valeur réelle telle que définie par le processus. Le programme principal demandera à l'utilisateur le nombre qu'il choisit, appellera la fonction *calculer* en passant la valeur saisie en paramètre, et affichera le résultat. Traduire les algorithmes en Python

Algorithme et code de la fonction *carre*

Voici l'algorithme de la fonction *carre*, comme vu en courss.

Algorithme 1 la fonction carré

1: **fonction** CARRE(x : *réel*) : réel
2: **variables locales :** ▷ **aucune variable locale**
3: **renvoyer** $x * x$
4: **Fin fonction**

Voici la traduction de cet algorithme en Python :

```
def square(a:float)->float:
    return a*a
```

Exercice 2

Rappel sur la division des entiers (division euclidienne) : 15 divisé par 2 donne 7 reste 1. En algorithmique (et en Python) nous avons deux opérateurs : pour le reste de la division *mod (%)* et le quotient de la division euclidienne *div (//)*.

Ainsi :

— 15 *div* 2 donne 7,
— 15 *mod* 2 donne 1.

Question A

Écrire l'algorithme, puis le code Python, d'une fonction qui à partir de deux entiers (a et b) passés en paramètres (strictement positifs et formés chacun de deux chiffres) retourne un entier de 4 chiffres tel que le nombre b est intercalé entre les deux chiffres de a. Exemple : Paramètres de la fonction : a=56, b=21 Résultat : 5216

Question B

Écrire l'algorithme, puis le code Python, d'une fonction (saisieInt) qui admet 2 valeurs entières en paramètre (min et max), la fonction demandera à l'utilisateur de saisir une valeur entière comprise entre min et max, tant que la valeur saisie est erronée la fonction redemande la saisie. Quand la valeur saisie est correcte elle est renvoyée par la fonction.

Question C

Écrire le programme principal qui demande à l'utilisateur de saisir deux entiers compris dans l'intervalle [10 , 99] et qui affiche le nombre retourné par la fonction de la question précédente avec en paramètres les valeurs saisies.

Question D

Modifier le programme principal pour effectuer la saisie des deux nombres avec la fonction *saisieInt*

Fiche de TD 2

Cette fiche a pour but de travailler sur la notion de branchement conditionnel (*Si Alors Sinon*).

Exercice 1

Écrire l'algorithme et le code Python d'une fonction retournant la valeur absolue de la valeur réelle passée en paramètre.

Exercice 2

Solveur d'équations.

Question A

On considère une équation du premier degré : $a \cdot x + b = 0$. Écrire l'algorithme et le code Python d'une fonction retournant la solution d'une équation dont les coefficients sont passés en paramètre.

Question B

Une équation du second degré s'écrit : $a \cdot x^2 + b \cdot x + c = 0$. Dans la bibliothèque *math* sont définies deux fonctions :

— *sqrt(x :float)->float* : retourne la racine carrée (square root) de la valeur (x) passée en paramètre ;

— *pow(x :float, n :float)->float* : retourne x^n

Pour que votre script Python puisse utiliser ces deux fonctions il faut mettre au début du fichier la ligne suivante :

```
from math import sqrt,pow
```

Pouvez-vous écrire une fonction permettant de résoudre dans $\mathbb{R}$ une équation du second degré ?

Question C

Écrire l'algorithme et le code Python d'une fonction retournant le nombre de solutions dans $\mathbb{R}$ d'une équation du second degré dont les coefficients sont passés en paramètre.

Question D

Écrire l'algorithme et le code Python du programme principal qui demandera à l'utilisateur les valeurs des coefficients de l'équation, puis qui affichera le nombre de solutions ainsi que leur valeurs.

Exercice 3

On peut représenter un triangle par les longueurs de ses côtés.

Question A

Écrire l'algorithme et le code Python d'une fonction qui admet en paramètre 3 réels correspondant aux longueurs des côtés d'un triangle et qui retourne vrai si le triangle est équilatéral.

Question B

Écrire l'algorithme et le code Python d'une fonction qui admet en paramètre 3 réels correspondant aux longueurs des côtés d'un triangle et qui retourne vrai si le triangle est isocèle.

Question C

Écrire l'algorithme et le code Python d'une fonction qui admet en paramètre 3 réels correspondant aux longueurs des côtés d'un triangle et qui retourne vrai si le triangle est rectangle.

Question D

Écrire l'algorithme et le code Python d'un programme principal qui demande la longueur des côtés d'un triangle et qui affiche la nature du triangle.

Fiche de TD 3

L'objectif de cette fiche est de travailler sur les boucles.

Exercice 1

Écrire un programme qui permet d'afficher la table des multiplications comme ci-dessous :

```
X   0   1   2   3   4   5   6   7   8   9   10
0   0   0   0   0   0   0   0   0   0   0   0
1   0   1   2   3   4   5   6   7   8   9   10
2   0   2   4   6   8   10  12  14  16  18  20
3   0   3   6   9   12  15  18  21  24  27  30
4   0   4   8   12  16  20  24  28  32  36  40
5   0   5   10  15  20  25  30  35  40  45  50
6   0   6   12  18  24  30  36  42  48  54  60
7   0   7   14  21  28  35  42  49  56  63  70
8   0   8   16  24  32  40  48  56  64  72  80
9   0   9   18  27  36  45  54  63  72  81  90
10  0   10  20  30  40  50  60  70  80  90  100
```

Exercice 2

Écrire une fonction qui admet un entier en paramètre (c). Le paramètre c déterminera le nombre de lignes. L'affichage alternera les caractères * et . comme illustré ci-dessous ($c = 6$) :

```
******
*****.
****..
***...
**....
*.....
```

Fiche de TD 4

Cette fiche a pour objectif de travailler la notion de récursivité et d'utiliser des listes

Exercice 1

On se propose d'étudier différents algorithmes permettant de calculer x^n, $n \in \mathbb{N}$

Question A

Proposer un algorithme évident en itératif et en récursif. Donner les codes Python.

Question B

Sachant que $x^{2n} = (x^n)^2$ et que $x^{2n+1} = x.(x^n)^2$, proposer un nouvel algorithme. Selon vous quel est le plus performant ? pourquoi ?

Exercice 2

Un nombre naturel (entier positif) est dit premier si et seulement si il n'est divisible que par 1 et lui-même. Ainsi, le premier nombre premier est 2 le suivant est 3 puis 5.

Question A

Comment déterminer algorithmiquement que b divise a ?

Question B

Donner une version itérative et une version récursive d'un algorithme qui admet, en paramètre, un entier et qui retourne vrai ou faux selon que l'entier est un nombre premier ou non. Donner le code Python de ces deux algorithmes.

Question C

Donner un algorithme et le code Python qui retourne une liste contenant les n premiers nombres premiers (n étant passé en paramètre).

Question D

Donner un algorithme et le code Python qui retourne une liste contenant les n premiers nombres premiers (n étant passé en paramètre) exploitant le principe de récursivité terminale (les resultats intermédiaires sont passés en paramètre). Quel est l'intérêt de cette approche ?

Exercice 3

La suite de Fibonacci est définie comme suit :
$F_n = F_{n-1} + F_{n-2}$
$F_0 = 0$
$F_1 = 1$

Question A

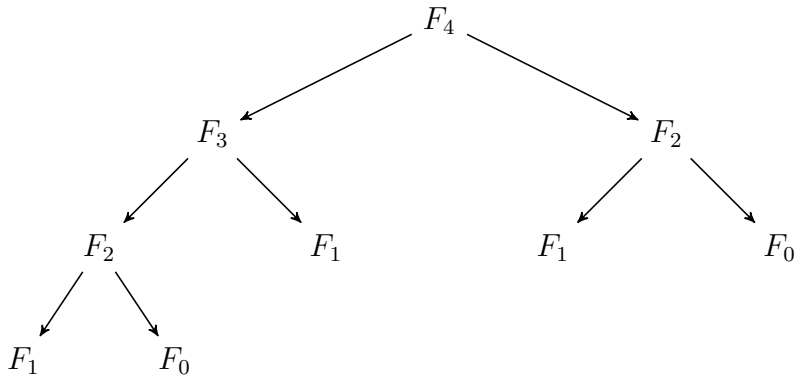
Donner un algorithme itératif et un récursif ainsi que les codes Python d'une fonction calculant F_n, n étant passé en paramètre.

Question B

Combien d'appels, en récursif, sont réalisés pour calculer F_5 ? Combien d'appels, en récursif, sont réalisés pour calculer F_6 ?

Question C

Pour calculer F_4 on a l'arbre d'appels suivant :



On appelle donc plusieurs fois les mêmes calculs (3 fois celui de F_1, 2 fois celui de F_0 et F_2) Proposer un algorithme récursif terminal permettant de calculer F_n sans lancer plusieurs fois les mêmes calculs (on pourra construire la liste des nombres de Fibonacci et passer cette liste comme résultat intermédiaire).

Fiche de TD 5

Cette fiche a pour objectif de travailler sur les listes.

Exercice 1

On suppose travailler avec des listes de réels.

Question A

Écrire un algorithme et le code Python retournant l'indice de la première occurrence d'une valeur dans une liste. Si la valeur n'est pas présente, la fonction retournera -1.

Question B

On suppose maintenant que la liste est triée. Proposer un algorithme tenant compte de ce fait.

Question C

Proposer un algorithme de recherche dichotomique. D'après Wikipédia :

> "*La recherche dichotomique, ou recherche par dichotomie1 (en anglais : binary search), est un algorithme de recherche pour trouver la position d'un élément dans un tableau trié. Le principe est le suivant : comparer l'élément avec la valeur de la case au milieu du*

tableau ; si les valeurs sont égales, la tâche est accomplie, sinon on recommence dans la moitié du tableau pertinente."

Comme pour les algorithmes précédents : la fonction retournera l'indice de la première occurrence d'une valeur dans une liste ; si la valeur n'est pas présente, la fonction retournera -1.

Exercice 2

On suppose travailler avec des listes de réels.

Question A

Écrire un algorithme et le code Python d'une fonction calculant le nombre d'occurrence d'une valeur dans une liste.

Question B

Même question, mais en sachant que la liste est triée.

Question C

Écrire un algorithme et le code Python d'une fonction retournant le minimum et le maximum d'une liste.

Question D

Même question, mais en sachant que la liste est triée.

Exercice 3

On suppose travailler avec des listes de réels.

Question A

Écrire un algorithme et le code Python d'une fonction calculant la valeur moyenne d'une liste.

Question B

Écrire un algorithme et le code Python d'une fonction admettant deux listes de même taille et qui retourne une liste dont les valeurs correspondent aux sommes élément par élément des deux listes passées en paramètres d'entrée.

Question C

Écrire un algorithme et le code Python d'une fonction admettant une liste en paramètre et qui renvoie une liste contenant les valeurs de cette liste **sans doublon**.

Question D

Écrire un algorithme et le code Python d'une fonction admettant deux listes qui retourne une liste dont les valeurs correspondent aux éléments communs aux deux listes passées en paramètres d'entrée.

Question E

Écrire un algorithme et le code Python d'une fonction admettant deux listes qui retourne une liste dont les valeurs correspondent aux éléments distincts aux deux listes passées en paramètres d'entrée.

Fiche de TD 6

Cette fiche a pour objectif de travailler sur les dictionnaires.

Exercice 1

On suppose qu'un enseignant a réalisé une évaluation. Le secrétariat lui a transmis une liste dont chaque élément est un dictionnaire comportant deux clés : "nom" et "prénom".

Question A

Écrire une fonction qui admet la liste des nom et prénom de chaque étudiant et qui demande à l'enseignant la note obtenue et l'ajoute (on choisira comme clé "note").

Question B

Écrire une fonction qui retourne le nom et prenom de ou des étudiant(s) qui a (ont) obtenu la meilleure note.

Question C

Écrire une fonction qui admet la liste des notes (comprises entre 0 et 20) en paramètre d'entrée et qui retourne une liste de fréquence des notes. Ainsi, $l[n]$ contient le nombre d'étudiants ayant obtenu une note comprise

dans l'intervalle $[n, n+1[$. Par exemple, $l[0]$ contient le nombre d'étudiants ayant obtenu une note inférieure à 1.

Fiche de TD 7

Cette fiche a pour objectif de travailler sur les chaînes de caractères

Exercice 1

Compléments de cours :

- `codeAscii = ord('A')` met la valeur 65 dans *codeAscii*.
- `lettre = chr(65)` met la valeur 'A' dans *lettre*.
- Extraction de sous-chaînes (ou de sous collections, ce qui est vrai pour les **str** l'est aussi pour les **list** et les **tuple**) :
  - soit `x = 'abcdef'` : définition de la chaîne.
  - `print(x[2])` affiche : c, le 3ème caractère (indice commence à 0).
  - `print(x[0:3])` affiche abc : les caractères d'indices 0 à 2(3-1).
  - `print(x[1:])` affiche bcdef : les caractères à partir de l'indice 1.
  - `print(x[:3])` : affiche abc : les caractères jusqu'à l'indice 2 (3-1).
  - `print(x[-2:])` : affiche ef : les 2 derniers caractères.
  - `print(x[0:-2])` : affiche abcd : toute la chaîne sans les 2 derniers caractères. Si l'index de fin est > à la longueur, c'est la longueur qui est utilisée.
  - `print(x[::2])` affiche ace : chaîne avec un caractère sur deux (en commençant par le premier).
  - `print(x[1::2])` affiche bdf : chaîne avec un caractère sur deux en commençant par le deuxième.
  - `print(x[::-1])` affiche fedcba : la chaîne renversée.
  - `print(list(x))` affiche ['a', 'b', 'c', 'd', 'e', 'f'] : une liste des caractères de la chaîne.

Question A

Un palindrome est un mot qui peut se lire dans les deux sens : par exemple, KAYAK est un palindrome. Écrire une fonction qui admet en paramètre une chaîne de caractères et qui retourne un booléen valant vrai si la chaîne correspond à un palindrome.

Question B

Écrire une fonction qui admet deux chaînes de caractères en paramètre (s1 et s2) et qui retourne le nombre de fois où s2 est présent dans s1.

Question C

Écrire une fonction qui admet en paramètre une chaîne de caractères et qui retourne le nombre de voyelles contenue dans la chaîne.

Question D

Écrire une fonction qui admet en paramètre une chaîne de caractères et qui retourne une chaîne de caractères identique mais avec les majuscules converties en minuscules.

Question E

Écrire une fonction qui admet deux chaînes de caractères en paramètre (s1 et s2) et qui retourne vrai si s2 est une anagramme de s1.

Question F

Écrire une fonction qui admet deux chaînes de caractères en paramètre (s1 et s2) et qui retourne le nombre de lettres communes aux deux chaînes.

Fiche de TD 8

Cette fiche a pour objectif de travailler sur les matrices représentées sous forme de listes.

Une matrice peut être considérée comme une liste de listes (correspondant aux lignes) de même taille. Ainsi une matrice 3x2 (3 lignes, 2 colonnes) sera représentée par une liste de 3 listes , chacune possédant exactement 2 éléments.

Soit A, une matrice, $a_{i,j}$ correspond à la valeur située à l'intersection de la ligne i et de la colonne j de la matrice A. On notera $A = (a_{i,j})$ et on dira que A est de dimension (m, n) si elle possède m lignes et n colonnes.

Exercice 1

Question A

Écrire une fonction en Python qui retourne une liste dont les valeurs sont initialisées à 0 et qui correspond à une matrice dont on aura passé en paramètre le nombre de lignes et de colonnes.

Question B

Écrire une fonction en Python qui retourne un booléen selon qu'une valeur, passée en paramètre, est présente ou non dans une matrice, passée également en paramètre.

Question C

Écrire une fonction en Python qui retourne la somme de 2 matrices passées en paramètres. Soient 2 matrices, toutes deux de dimension (m, n), $A = (a_{i,j})$ et $B = (b_{i,j})$.
Soit C, la somme de A et B notée $C = A + B$. On a $C = (c_{i,j})$ est de dimension (m, n) et est définie de la manière suivante :
$c_{i,j} = a_{i,j} + b_{i,j}$

Question D

Écrire une fonction en Python qui retourne le produit d'un scalaire (un réel) avec une matrice passés en paramètres. Soient a un scalaire et B une matrice, $B = (b_{i,j})$, de dimension (m, n), et le produit de a et B noté $C = a.B$. On a $C = (c_{i,j})$ est de dimension (m, p) et est définie de la manière suivante :
$c_{i,j} = a.b_{i,j}$

Question E

Écrire une fonction en Python qui retourne le produit de 2 matrices passées en paramètres. Soient 2 matrices, $A = (a_{i,j})$, de dimension, (m, n), et $B = (b_{i,j})$, de type (n, p).
Soit C, le produit de A et B noté $C = AB$. On a $C = (c_{i,j})$ est de dimension (m, p) et est définie de la manière suivante :
$c_{i,j} = \sum_{k=0}^{n} a_{i,k}.b_{k,j}$

Question F

Écrire une fonction en Python qui retourne la transposée d'une matrice passée en paramètre. Soit une matrice, $A = (a_{i,j})$, de dimension (m, n). A^T est la transposée de A. A^T est de dimension (n, m) et est définie de la manière suivante : $A^T = (a_{j,i})$

Exercice 2

Soit le système d'équations suivant :

$$\begin{cases} a_{0,0}.x_0 + ... + a_{0,n}.x_n &= b_0 \\ . &= . \\ . &= . \\ . &= . \\ a_{n,0}.x_0 + ... + a_{n,n}.x_n &= b_n \end{cases}$$

On peut l'écrire sous la forme matricielle suivante :
$A.X = B$,
où $A = (a_{i,j})$ est la matrice des coefficients de dimension (n, n) (A est une matrice carrée, il faut autant d'équations qu'on a de variables), $X = (x_i)$ est le vecteur des n variables (représenté comme une matrice de dimension (1,n)) et $B = (b_i)$ est le vecteur des n résultats

L'objectif de cet exercice est de réaliser l'algorithme de résolution d'un système d'équations. Pour la suite on modélise un système d'équation avec les deux matrice A et B, l'objectif étant de trouver la matrice X.

Question A

On peut réaliser des opérations (combinaisons linéaires) sur les lignes. Soit L_i et L_j deux lignes du système, on peut réaliser l'opération suivante : $L_j = L_i + k.L_j$, où k est un réel. Écrire une fonction qui admet la matrice des coefficients et la matrice des résultats en paramètre ainsi que 2 entiers (i et j) et un réel k la fonction modifiera les deux matrices afin de réaliser la combinaison linéaire : $L_j = L_i + k.L_j$

Question B

Soit le système d'équations suivant :

$$\begin{cases} 2.x_0 - 3.x_1 + x_2 &= 6 \\ 0.x_0 + 2.x_1 + 3.x_2 &= 9 \\ 0.x_0 + 0.x_1 + x_2 &= 1 \end{cases}$$

Le système est "triangularisé". Sa résolution peut se faire "facilement" en remontant de ligne en ligne. A partir de la dernière ligne on sait que : $x_2 = 1$, on remplace dans la ligne au-dessus. On en déduit que $x_1 = 3$, et on continue

la remontée et la substitution des variables x_i par leur valeur et on obtient $x_0 = 7$.

Proposer un algorithme, ainsi que le code Python, d'une fonction permettant, à partir de combinaisons linéaires (fonction réalisée à la question de précédente) de trianguliariser le système.

Proposer ensuite un algorithme et sa traduction en Python d'une fonction permettant de résoudre un système triangularisé. La fonction retournera une matrice d'autant de ligne que de variables, chaque ligne n'ayant qu'une valeur, celle de x_i.

Pour vérifier que votre résolution est bonne si X est votre matrice solution obtenue, A, la matrice des coefficients du système au départ, et B la matrice des résultats au départ, alors, si vous calculez $A.X$ vous devez obtenir B

Fiche de TD 9

Nous allons étudier dans cette fiche le tri fusion. Il s'agit d'un tri dont la complexité est d'ordre $n.log(n)$.

Le principe est simple, imaginons qu'une liste (l) soit composée de 2 sous-listes triées (sl_1 et sl_2) : l = [2,4,5,8,1,3,9,12,15], sl_1 = [2,4,5,8] et sl_2=[1,3,9,12,15] Il est alors simple de d'obtenir la liste $lfus$ telle qu'elle soit composée des éléments de sl_1 et sl_2 et que tous les éléments soient dans l'ordre. Pour ce faire on va remplir la liste $lfus$ itérativement, on associe un indice (j et k) pour les 2 sous-listes, à chaque itération on choisit le plus petit entre $sl_1[j]$ et $sl_2[k]$. Il faut ensuite recopier la liste $lfus$ dans l.

Le tri fusion se base sur ce principe : On considère une partition de la liste à trier (définie par 2 indices),

- si la partition l a moins de 2 éléments, alors on ne fait rien,
- s'il y a exactement 2 éléments on fait une comparaison et on permute éventuellement les éléments,
- sinon on décompose la partition en 2 partitions, on lance leur tri (sur le même principe) et on fusionne les deux partitions triées.

Exercice 1

Question A

Écrire une fonction *remplacer* qui remplace à partir d'un indice, les éléments d'une liste, par les éléments d'une autre liste.

Question B

Écrire une fonction *fusionner* qui fusionne dans une liste deux partitions adjacentes triées. En paramètres, on aura donc la liste, l'indice du premier élément de la première partition, l'indice du dernier élément de la première partition et l'indice du dernier élément de la deuxième partition.

Question C

Écrire une fonction qui trie une partition d'une liste.

Question D

Écrire une fonction qui trie une liste

Fiche de TD 10

Exercice 1

Question A

Écrire une fonction qui admet en paramètre une liste triée et une valeur à insérer dans la liste. La fonction devra insérer la valeur dans la liste de telle sorte que la liste reste triée.

Question B

Même question mais avec une liste chaînée (reprendre le code donné en cours).

Fiche de TD 11

Cette fiche de TD a pour but d'illustrer les éléments vus en cours sur les types structurés définis par des classes.

Exercice 1

On considère qu'on souhaite représenter un *Instant* par :

— l'heure,
— la minute et
— la seconde actuelles.

Dans cette fiche on considère que les différents *Instant* ont lieu dans la même journée.

Question A

Écrire en Python la classe *Instant*.

Question B

Écrire la fonction *read_Instant()->Instant* qui apres avoir demandé à l'utilisateur la valeur des heures, minutes et secondes renvoie un *Instant* initialisé avec les valeurs saisies.

Question C

Écrire la fonction *print_Instant(t :Instant)* qui admet en paramètre un instant et qui affiche ses valeurs de la forme *h :m :s*.

Question D

Écrire la fonction *estInstantValide(t :Instant)->bool* qui retourne vrai si l'instant passé en paramètre est valide et faux sinon.

Question E

Écrire la fonction *suivant(t :Instant)->Instant* qui admet en paramètre un *Instant* et qui retourne un *Instant* correspondant à l'instant passé en paramètre + 1 seconde.

Question F

Écrire la fonction *estPlusRecent(t1 :Instant,t2 :Instant)->bool* qui admet deux instants en paramètres et qui retourne vrai si t1 est plus récent que t2 et faux sinon.

Question G

Écrire la fonction *duree(debut :Instant ,fin :Instant)->Instant* qui admet en paramètres deux instant et qui retourne un *Instant* correspondant à la durée entre fin et debut.

Fiche de TD 12

L'objectif de cette fiche est de travailler la notion de Pile.

Exercice 1

Question A

Écrire une fonction qui admet en paramètres 2 nombres et une chaîne de caractères. La chaîne de caractères sera une parmi les suivantes : "+","-","*","/" et correspondra à un opérateur arithmétique. La fonction retournera le résultat de l'opération.

Question B

On suppose avoir la classe Pile telle que définie dans le cours. Même question que la précédente mais en paramètre on a maintenant un Pile dans laquelle on a empilé les deux nombres puis l'opérateur.

Question C

Même question mais maintenant on a une Pile dans laquelle on a empilé l'opérateur puis les deux nombres.

Question D

Proposer comment empiler les nombres et les opérateur ainsi que l'algorithme de calcul pour l'opération suivante : 6+3-4

Question E

Même question pour l'opération suivante : (5-2)*3

Fiche de TD 13

Exercice 1

Définir la classe Point en Python. Définir la classe Polygone en Python

Question A

Écrire une fonction qui admet 2 Point en paramètre et qui retourne la distance séparant ces 2 Point.

Question B

Écrire une fonction qui admet en paramètre un Polygone et qui retourne le périmètre de ce Polygone.

Exercice 2

Un Polynome de degré n ($a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0$) peut être représenté par la liste des coefficients a_i associés à chaque puissance de x. Définir la classe Polynome en Python.

Question A

Écrire une fonction qui admet un Polynome et une liste de réels en paramètre et qui permet de modifier le Polynome pour y associé aux coefficients la liste passée.

Question B

Écrire une fonction qui admet un Polynome P et un réel v en paramètre et qui retourne la valeur de $P(v)$.

Question C

Écrire une fonction qui admet un Polynome P en paramètre et qui retourne le Polynome dérivé.

Fiche de TD 14

On se propose de simuler un jeu de Bataille (règle simplifiée). On distribue aléatoirement à deux joueurs 26 cartes chacun. A chaque tour, les deux joueurs tirent la première carte de leur pile. Si les cartes sont de même valeur elles sont placées dans un tas. Sinon le joueur qui a retourné la carte de plus grande valeur récupère les deux cartes et celles du tas de bataille dans son tas de plis. Quand un joueur a épuisé sa pile de cartes : — s'il n'a pas de cartes dans son tas de plis, alors il a perdu, — sinon, il mélange son tas de plis qui devient sa nouvelle pile de cartes On se propose d'utiliser : LesCouleurs = ["Coeur","Carreau","Pique","Treffle"] LesValeurs = { "2" :1,"3" :2,"4" :3,"5" :4,"6" :5,"7" :6,"8" :7,"9" :8, "10" :9,"Valet" :10,"Dame" :11,"Roi" :12,"As" :13 } Un Joueur possède un nom.

Exercice 1

Écrire les classes Carte et Joueur

Exercice 2

Une Partie possède 2 Joueurs et un tas vide de carte issue des batailles. Écrire la classe Partie.

Exercice 3

Écrire une fonction ($creerJdC()$) qui retourne une liste de 52 cartes.

Exercice 4

Écrire une fonction ($melanger(j)$) qui mélange le jeu de carte, j, passé en paramètre.

Exercice 5

Écrire une fonction, distribuer(p, j), qui permet de distribuer les cartes du jeu, j, aux joueurs de la partie, p.

Exercice 6

Écrire une fonction, $comparer(a, b)$, qui retourne 1 si la carte a est de plus grande valeur que b, 2, si c'est l'inverse et 0 si elles sont de même valeur.

Exercice 7

Écrire une fonction, $jouerTour(p)$, qui admet une Partie, p, en paramètre et qui permet de jouer un tour (chaque joueur retourne la première carte de sa pile, et les cartes sont comparées et gagnées par un des joueur ou mis sur le tas de la partie) le résultat du tour est affiché.

Exercice 8

Écrire une fonction, $lancerPartie()$, qui permet de créer une $Partie$ et visualiser les information du déroulement de la bataille :

— demande le nom des joueurs,
— crée,mélange et distribue un jeu de 52 cartes,
— déroule les tours de jeu,
— félicite le gagnant.

Fiche de TD 15

Nous allons travailler sur les arbres binaires de recherche. Selon Wikipédia : *En informatique, un arbre binaire est une structure de données qui peut se représenter sous la forme d'une hiérarchie dont chaque élément est appelé nœud, le nœud initial étant appelé racine. Dans un arbre binaire, chaque élément possède au plus deux éléments fils au niveau inférieur, habituellement appelés gauche et droit. Du point de vue de ces éléments fils, l'élément dont ils sont issus au niveau supérieur est appelé père. Au niveau le plus élevé il y a donc un nœud racine. Au niveau directement inférieur, il y a au plus deux nœuds fils. En continuant à descendre aux niveaux inférieurs, on peut en avoir quatre, puis huit, seize, etc. c'est-à-dire la suite des puissances de deux. Un nœud n'ayant aucun fils est appelé feuille. Le nombre de niveaux total, autrement dit la distance entre la feuille la plus éloignée et la racine, est appelé hauteur de l'arbre. Le niveau d'un nœud est appelé profondeur. Les arbres binaires peuvent notamment être utilisés en tant qu'arbre binaire de recherche* Un arbre binaire de recherche est un arbre binaire pour lequel le fils gauche d'un noeud contient une valeur plus petite et et à droite une valeur plus grande. La figure suivante montre une représentation schématique d'un arbre binaire de recherche (Cf. figure 15.1).

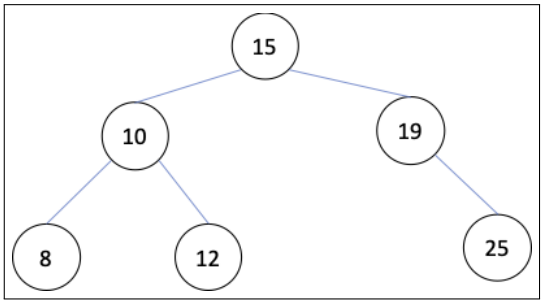


FIGURE 15.1 – Schéma illustrant un Arbre Binaire de Recherche.

Exercice 1

Proposer les classes permettant de gérer un Arbre Binaire de Recherche.

Exercice 2

Écrire une fonction permettant d'ajouter un élément dans un Arbre Binaire de Recherche

Exercice 3

Écrire une fonction permettant d'afficher (dans l'ordre croissant) les valeurs stockées dans un Arbre Binaire de Recherche.

Exercice 4

Écrire une fonction retournant la hauteur d'un Arbre Binaire de Recherche.