A] Premier pas (chercher parfois sur le web)

Exercice 1: **fonctions simples** Ecrire les fonctions permettant de:

-Calculer le résultat d'une somme, d'un produit, d'une différence

-Calculer le quotient de la division euclidienne

-Calculer le reste de la division euclidienne

- calculer le résultat de la division réelle de deux nombres

-tester l'égalité entre deux calculs (par exemple 6 et 2\*3)

B] Fonctions simples fichier TP4\_fonctionSimple\_ \_NomEleve

Exercice 3: **Fonction mystere** soit le code suivant:

*def mystere(a,b,c):*

*if a<c:*

*if a<b:*

*return a*

*else:*

*return b*

*else:*

*if b<c:*

*return b*

*else:*

*return c*

1) Quel est le résultat de l'application de mystere(1.0,3.5,5.9) ?

2) Que fait la fonction ? renommer la fonction et lui donner une entête correcte.

3) Combien de tests différents faut il faire pour s'assurer que la fonction est correcte.

exercice 4 : **Fonction de conversion**

-Ecrire une fonction farhenheitEnCelsius qui convertit une temperature donnée en farhenheit en celsius (c=5\*(f–32)/9 ).Reflechir aux types des arguments et de la valeur de retour.

-proposer une fonction qui réalise la conversion inverse de la précédente.

En résumé

1) Ecrire l'entête de la fonction

2) Trouver l'algorithme pour résoudre le problème (trés simple pour la 1ere partie puisque la solution est donnée. Il faut simplement résoudre l'équation en f pour la 2eme)

3) Traduire l'algorithme dans le langage

4) Tester la fonction.

Exercice 5: **Entier Naturel ?**

Ecrire une fonction *Naturel* qui teste si un entier est un entier naturel ou non.

Exercice 6 **Divisible ?**

Ecrire une fonction *Divisible* qui, étant donnés deux nombres entiers, teste si le premier est divisible par le second.

(Manipulation des opérareur %, / et // pour la division euclidienne.)

Exercice 7 - **Nombre premier**

A l'aide de la fonction précédente, écrire une fonction nbPremier permettant de déterminer si un nombre donné est premier ou non.

Exercice 8-**Trouver l'erreur**

Dans ce qui suit, il est demandé de prédire ce que vont faire les programmes avant de les tester.

* Soit le code suivant:

*def malade(x):*

*if (x=37.5):*

*return False*

*else:*

*return True*

Que va-t-il se passer lors de l'appel de *malade(39.0)* ? corriger le code.

* Soit le code suivant:

*x=float(4/3)*

*print(x)*

Qu'affiche ce bout de programme ? Pourquoi ?

* Soit le code suivant:

*if 3<5:*

*x=6/2*

*else:*

*x=6/0*

*print x*

Que fait le programme lors de l'exécution de ce code ?

* Soit maintenant la fonction:

*def si(b,e1,e2):*

*if b:*

*return e1*

*else:*

*return e2*

Que se passe-t-il lors de l'appel de fonction si(3<5,6/2,6/0)

Exercice 9-**Echange de valeurs**

Soit la fonction suivante:

*def echange(x,y):*

*tmp=x*

*x=y*

*y=tmp*

Que vaut la variable a après l'exécution suivante:

*a=1*

*b=5*

*echange(a,b)*

*print a,b*

Soit la fonction suivante:

*def echange2(A):*

*tmp=A[0]*

*A[O]=A[1]*

*A[1]=tmp*

Que vaut la variable M après l'exécution suivante:

*a=1*

*b=5*

*M=[a,b]*

*print M*

*echange2(M)*

*print M*

C] Les fonctions récursives (Avancé)

fichier TP4\_fonctionAvance\_ NomEleve

Exercice 10 -**Somme des n premiers entiers**

Ecrire une fonction *sommeN* qui, étant donné un entier n, renvoi la valeur de .

Vérifier que cette somme est bien toujours égal à n(n+1)/2.

Vous chercherez des informations sur ce qu'est une fonction récursive puis essayerez de programmer la même fonction de façon récursive ;que vous nommerez *sommeNrecursive.*

Exercice 11-**Calcul du PGCD**

On rappelle que l'algorithme d'Euclide permet de déterminer le PGCD de deux entiers à l'aide des division s euclidiennes en remarquant que:

-pgcd(n1,n2)=n1 si n2=0

-pgcd(n1,n2)=pgcd(n2,r) sinon où r est le reste de la division euclidienne de n1 par n2 (on suppose n2 ≤ n1)

A partir de ce principe, écrire une fonction récursive *pgcd* calculant le PGCD de deux entiers. En déduire une fonction *premierEntreEux* testant si deux entiers sont premiers entre eux.

Exercice 12- **Puissance**

Ecrire une fonction récursive *puissance* qui, étant donné un nombre x et un entier n, renvoie la valeur de xn.

Ecrire également sa variante itérative (Avec une boucle)

exercice 14 - **Approximation de la racine carrée.**

Le but de cet exercice est d'écrire une fonction qui calcule une valeur approchée de la racine carrée d'un nombre. Soit x un nombre réel positif, une valeur approchée de est donnée par le calcul des valeurs de la suite suivante:

Ecrire une fonction qui, étant donne un nombre x et un entier n, renvoie l'approximation au rang n de .

Exercice 15-**Fibonnacci**

Ecrire une fonction récursive *fibonacci* qui, étant donné un entier n, renvoie la valeur de la suite de Fibonacci au rang n:

Donner la variante itérative (qui n'est pas récursive). Que pensez vous de la complexité des deux variantes ?