

	<i>Université de Corse - Pasquale PAOLI</i>	
	Diplôme : Licence SPI 3^{ème} année	2023-2024
	UE : Ateliers de programmation	
	Atelier 1 : Fonctions et paramètres - Fonctions renvoyant un résultat - Fonctions paramétrées	
	Enseignants : Paul-Antoine BISGAMBIGLIA, Marie-Laure NIVET, Evelyne VITTORI	

EXERCICE 1 - Fonction chaîne de caractère : Calcul d'IMC

1. Définissez une fonction `message_imc` qui admet en paramètre un nombre réel représentant l'indice de masse corporelle (IMC) d'une personne et renvoie une chaîne de caractère correspondant à l'interprétation de cet imc selon le tableau ci-dessous.

IMC	Interprétation
< 16,5	dénutrition ou famine
16,5 à 18,5	maigre
18,5 à 25	corpulence normale
25 à 30	surpoids
30 à 35	obésité modérée
35 à 40	obésité sévère
plus de 40	obésité morbide

2. Définissez une fonction de test réalisant plusieurs appels successifs de votre fonction `message_imc`.
3. Lancez vos tests en invoquant votre fonction de test

EXERCICE 2 - Fonction booléenne : Année bissextile

Une année est bissextile si elle est divisible, par 4 et non par 100, ou par 400.

Définissez une fonction `est_bissextile` qui prend en paramètre un nombre entier représentant une année et renvoie un booléen indiquant si l'année considérée est bissextile ou non.

Indication : Pour savoir si un nombre x est divisible par 4 (par ex), il suffit de tester que le reste de sa division par 4 est égal à 0. L'opérateur modulo (noté `%` en python) fait cette opération. Ainsi si $x \% 4 == 0$, cela signifiera que x est divisible par 4.

Comme pour l'exercice 1 définissez une fonction de test contenant des tests pertinents.

EXERCICE 3 - Résolution d'une équation du second degré ax^2+bx+c

On souhaite écrire un programme qui permette la résolution d'une équation du second degré de la forme $ax^2+bx+c=0$ avec a,b,c réels et $a \neq 0$.
Les solutions de cette équation s'appellent des racines.

1. Définissez une fonction **discriminant(a,b,c)** qui admet en paramètre trois réels a,b et c et renvoie le discriminant calculé $\Delta = b^2 - 4ac$.
2. Définissez une fonction **racine_unique(a,b)** qui admet en paramètre deux réels a et b et renvoie la valeur $x = \frac{-b}{2a}$.
3. Définissez une fonction **racine_double(a,b,delta,num)** qui admet en paramètre deux réels a et b, un réel delta représentant le discriminant de l'équation et un entier num représentant le numéro de la racine. Si num=1, la fonction renvoie la valeur de la racine (solution) numéro 1 $x_1 = \frac{-b+\sqrt{\Delta}}{2a}$ et si num=2, la valeur de la racine numéro 2 $x_2 = \frac{-b-\sqrt{\Delta}}{2a}$.
4. Définissez une fonction **str_equation(a,b,c)** qui admet en paramètre trois réels a,b et c et renvoie une chaîne de caractères représentant la chaîne " $ax^2 + bx + c = 0$ " (format d'affichage)

Exemples

str(equation(2,3,4)) renvoie la chaîne $2x^2 + 3x + 4 = 0$
 str(equation(1,0,4)) renvoie la chaîne $x^2 + 4 = 0$
 str(equation(-1,1,-4)) renvoie la chaîne $-x^2 + x - 4 = 0$

5. Définissez une fonction **solution_equation(a,b,c)** qui admet en paramètre trois réels a,b et c et renvoie une chaîne de caractères représentant un message de la forme:
 - "Solution de l'équation $ax^2+bx+c=0$
Pas de racine réelle" si l'équation n'a pas de solution réelle
 - "Solution de l'équation $ax^2+bx+c=0$
Racine unique : $x = \dots$
 - "Solution de l'équation $ax^2+bx+c=0$
Deux racines :
 $x_1 = \dots$
 $x_2 = \dots$ "
6. Définissez une procédure **equation(a,b,c)** qui admet en paramètre trois entiers a,b et c et affiche la ou les solutions de l'équation ax^2+bx+c ou le message "aucune solution réelle".
7. Définissez une fonction de test réalisant plusieurs appels successifs de votre fonction équation correspondant à des jeux de test pertinents

Principe de résolution de l'équation $ax^2+bx+c=0$

Calcul du discriminant. $\Delta = b^2 - 4ac$

Si $\Delta < 0$ pas de racine réelle

Si $\Delta = 0$, une racine unique $x = \frac{-b}{2a}$

Si $\Delta > 0$, deux racines, $x_1 = \frac{-b+\sqrt{\Delta}}{2a}$ et $x_2 = \frac{-b-\sqrt{\Delta}}{2a}$

Attention : le discriminant ne devra être calculé qu'une seule fois.

EXERCICE 4 - Calculs de date

1. Définissez une fonction **date_est_valide(jour,mois,annee)** qui admet en paramètre trois entiers jour, mois et année et renvoie un booléen indiquant si la date est valide.

Nota-bene : utilisez la fonction `est_bissextile` de l'exercice 2

2. Définissez une fonction **saisie_date_naissance()** qui assure la saisie au clavier d'une date de naissance par la saisie de trois entiers année, mois et jour et renvoie une valeur de type date.
3. Définissez une fonction **age(date_naissance)** qui prend en paramètre une date de naissance et renvoie un entier correspondant à l'âge de la personne à la date du jour.
4. Définissez une fonction **est_majeur(date_naissance)** qui renvoie True si l'individu est majeur à la date du jour et False dans le cas contraire.
5. Définissez une procédure **test_acces ()** qui assure la saisie d'une date de naissance et affiche un message de la forme "Bonjour, vous avez .. ans, Accès autorisé " ou "Désolé, vous avez ..ans, Accès interdit" selon que l'individu est majeur ou non à la date du jour.
6. Testez vos différentes fonctions et procédures

Indications

- Les fonctions de manipulation de date sont incluses dans le module date du module datetime:

```
#à ajouter dans votre fichier de code
from datetime import date
```
- La date du jour est donnée par la fonction today :

```
aujourd'hui=date.today()
```
- L'instruction `date(2020, 9, 8)` renvoie un objet de type date correspondant à la date du 08/09/2020
- Les attributs year, month et day permettent de récupérer respectivement l'année, le mois et le jour d'une variable de type date :

```
annee=ma_date.year...
```