

1. Python

Dans ces travaux pratiques, vous allez apprendre l'algorithmie en python, on introduit les conditions

1.1. Géométrie

Q : Modifier le programme [aire_cercle.py](#) pour demander le type de calcul (sphère ou cercle) et rajouter le code du volume de la sphère, créer un fichier

1.2 Calculs

Q : Calcul du montant d'une remise

Un commerçant accorde une remise de 5 % pour tout achat d'un montant compris entre 100 et 500 € et 8 % au-delà. Écrire un programme [remise.py](#) de calcul du montant de la remise sur un achat donné.

Q : Année bisextile, porter l'algorithme suivant dans un programme [bisextile.py](#)

Si A n'est pas divisible par 4, l'année n'est pas bissextile.

Si A est divisible par 4, l'année est bissextile sauf si A est divisible par 100 et pas par 400.

Q : écrire un programme [second_degré.py](#) qui résout les équations du second degré du type : $ax^2 + bx + c = 0$

Il accepte a, b, c en argument. L'algorithme est expliqué ici :

<https://calculis.net/resoudre-equation-second-degre>

Note : ne pas calculer les nombres complexes si $\Delta < 0$ mais juste afficher que l'équation n'a pas de solution Réelle

1.3 Comparaison

Q : Ecrire un programme [dans_le_bon_ordre.py](#) faisant saisir trois entiers x, y, z à l'utilisateur, et lui indiquant si ces

nombres sont dans l'ordre croissant ($x \leq y \leq z$).

Q : Ecrire un programme [trier_3nombres.py](#) faisant saisir trois nombres réels x, y, z à l'utilisateur et qui les trie par ordre croissant (à la fin du déroulement du programme $x \leq y \leq z$).

Q : Ecrire un programme [signe.py](#) qui calcule le signe du produit de deux nombres sans calculer la valeur de ce produit. Par signe, on entend positif, négatif.

1.4 Sécurité, range check et tests

Q : reprendre le TP_06a, 2.3 [aire_cercle.py](#) et le modifier en [aire_cercle_controle.py](#) pour rajouter des contrôles pour vérifier que les valeurs saisies sont bien des nombres positifs non égaux à zéro

Q : range check TP_06a, 2.4 reprendre [tranche_chaine1.py](#) en [check_range.py](#) et avant de renvoyer les morceaux des chaînes, vérifier que les index de l'utilisateur sont bien compatibles avec la longueur de la chaîne