

TD1 : Contrôle du flux d'instructions

1. Saisissez un flottant. S'il est positif ou nul, affichez sa racine, sinon affichez un message d'erreur.

2. L'ordre lexicographique est celui du dictionnaire. Saisir deux mots, comparez-les pour trouver le « plus petit » et affichez le résultat. Refaire l'exercice en utilisant l'instruction ternaire :

`<res> = <a> if <condition> else `

3. On désire sécuriser une enceinte pressurisée.

On se fixe une pression seuil et un volume seuil : $p_{\text{Seuil}} = 2.3$,
 $v_{\text{Seuil}} = 7.41$.

On demande de saisir la pression et le volume courant de l'enceinte et d'écrire un script qui simule le comportement suivant :

– si le volume et la pression sont supérieurs aux seuils : arrêt immédiat;

– si seule la pression est supérieure à la pression seuil : demander d'augmenter le volume de l'enceinte ;

– si seul le volume est supérieur au volume seuil : demander de diminuer le volume de l'enceinte;

– sinon déclarer que « tout va bien ». Ce comportement sera implémenté par une alternative multiple.

4. Initialisez deux entiers : $a = 0$ et $b = 10$.

Écrire une boucle affichant et incrémentant la valeur de a tant qu'elle reste inférieure à celle de b . Écrire une autre boucle décrémentant la valeur de b et affichant sa valeur si elle est impaire. Boucler tant que b n'est pas nul.

5. Écrire une saisie filtrée d'un entier dans l'intervalle 1 à 10, bornes comprises. Affichez la saisie.

6. Affichez chaque caractère d'une chaîne en utilisant une boucle `for`. Affichez chaque élément d'une liste en utilisant une boucle `for`.

7. Affichez les entiers de 0 à 15 non compris, de trois en trois, en utilisant une boucle `for` et l'instruction `range()`.

8. Utilisez l'instruction `break` pour interrompre une boucle `for` d'affichage des entiers de 1 à 10 compris, lorsque la variable de boucle vaut 5.

9. Utilisez l'instruction `continue` pour modifier une boucle `for` d'affichage de tous entiers de 1 à 10 compris, sauf lorsque la variable de boucle vaut 5.

10. Utilisez une exception pour calculer, dans une boucle évoluant de -3 à 3 compris, la valeur de $\sin(x)/x$.

11. La clause `else` des boucles. Dans cet exercice, effectuez les saisies avec des `integerbox` et les affichages avec des `msgbox`, tous deux appartenant au module `easygui`. Initialisez une liste avec 5 entiers de votre choix puis saisissez un entier. Dans une boucle `for`, parcourez la liste. Si l'entier saisi appartient à la liste, sauvez-le et interrompez la boucle (puisque vous l'avez trouvé). Si la boucle s'est bien terminée, utilisez une clause `else` pour afficher un message l'annonçant. Entrez maintenant un autre entier, cette fois-ci positif. Écrivez une boucle `while` pour déterminer si cet entier est premier. S'il ne l'est pas, la boucle devra afficher le premier diviseur trouvé et s'interrompre. S'il est premier, l'afficher dans une clause `else`.