

Sujet n°2

Algorithmique & Programmation Problèmes algorithmiques itératifs

1 Premiers algorithmes

Exercice 1 : *Second degré*

Écrire un programme qui résout une équation du second degré.

Exercice 2 : *Surfaces et volumes*

QUESTION 1. Écrire un programme qui demande à l'utilisateur le rayon d'un disque et affiche en retour son périmètre et sa surface. Le calcul du périmètre et de la surface doivent se faire dans des fonctions séparées du reste du programme.

QUESTION 2. Écrire un autre programme qui demande à l'utilisateur le rayon d'un cylindre ainsi que sa hauteur et affiche sa surface et son volume. Vous devrez réutiliser les fonctions créées dans la question précédente.

Exercice 3 : *Tu devines mon nombre*

Écrire un programme qui génère un nombre au hasard entre 0 et 100 (voir fonction `random.randint`) et le fait deviner à un utilisateur. À chaque réponse de l'utilisateur, le programme indiquera si le nombre proposé est le bon ou s'il est inférieur ou supérieur au nombre à trouver.

Le programme doit boucler tant que la réponse n'a pas été trouvée.

Exercice 4 : *Je devine ton nombre*

Écrire un programme qui demande à un utilisateur de choisir un nombre qu'il doit garder secret, puis essayer de deviner ce nombre. À chaque proposition du programme l'utilisateur doit indiquer s'il s'agit du bon nombre ou s'il est inférieur ou supérieur au nombre à trouver.

Le programme doit boucler tant que la réponse n'a pas été trouvée, ou qu'une erreur de l'utilisateur a été détectée.

Exercice 5 : *Championnat*

Écrire un programme établissant un championnat type foot comportant N équipes. L'algorithme déterminant le $i^{\text{ème}}$ match d'une journée est le suivant :

Soient respectivement N , J et M , le nombre d'équipes, le numéro de la journée et le numéro du match de la journée.

Soit N_1 le nombre d'équipe à prendre en compte pour le calcul. Si N est pair : $N_1 = N$, sinon $N_1 = N + 1$. On aura alors $N_1 - 1$ jours de championnat et $\frac{N_1}{2}$ matchs par jours

Pour chaque jour et chaque match :

Équipe qui reçoit :

pour $M = 1$: N_1 si N est pair et 0 si N est impair (pas de match).
pour $M > 1$: $((J + M - 2) \% (N_1 - 1)) + 1$

Équipe en déplacement :
 $(J - M + N_1 - 1) \% (N_1 - 1) + 1$

Exercice 6 : Vérification de saisie

Écrire un programme qui demande à l'utilisateur de saisir un nombre et vérifie qu'il s'agit bien d'un nombre. S'il ne s'agit pas d'un nombre, afficher un message d'erreur et redemander à l'utilisateur une saisie ; sinon afficher le nombre.

Exercice 7 : Calculatrice

Écrire un programme calculatrice : saisir au clavier une opération (nombre opérande nombre) et afficher le résultat du calcul après avoir détecté la touche '='. Afficher un message d'erreur si besoin.

Ex : "3434+15=" (opérateurs +, -, *, /, %)

2 Les tableaux

Python ne possède pas par défaut de type tableau mais dispose d'un type liste. Pour les besoins des exercices suivants, vous utiliserez les listes de Python pour simuler des tableaux.

Exercice 8 : Saisie de liste

Créez un programme qui permet de remplir une liste avec des nombres qui seront saisis par l'utilisateur à l'aide du clavier. La saisie s'arrêtera lorsque l'utilisateur entrera une ligne vide.

Exercice 9 : Moyenne

Demander à l'utilisateur de saisir une liste de nombre, et afficher à la suite de chaque nombre fourni le minimum, le maximum et la moyenne des nombres précédemment saisis. Votre programme ne doit comporter qu'une seule boucle.

La saisie s'arrêtera lorsque l'utilisateur entrera une ligne vide.

Exercice 10 : Inversion

Demander à l'utilisateur de saisir une liste de nombre puis l'afficher. Inversez ensuite cette liste et l'afficher également.

Exercice 11 : Nombres pairs et impairs

QUESTION 1. Écrivez un programme qui crée une liste contenant des nombres compris entre 0 et 100, puis recopie tous les nombres pairs au début d'une nouvelle liste, et les nombres impairs à la fin de cette nouvelle liste.

QUESTION 2. Réécrivez votre programme sans utiliser de seconde liste.

Exercice 12 : Palindromes

À partir d'une liste de termes saisis par l'utilisateur, détecter et afficher uniquement les termes correspondant à un palindrome (mot pouvant se lire dans les deux sens ; ex : laval).

Exercice 13 : Tri à bulles

Créez un programme qui effectue un tri à bulles sur une liste de nombres.