Programmation 2 Spring, 2026

TD 3: loops

1. Écrivez un programme Python qui, étant donné un entier positif `n`, imprime tous les nombres dans $\{1, \ldots, n\}$.
2. Écrivez un programme Python qui imprime les nombres dans $\{0, \ldots, 100\}$ qui sont divisibles par 5 mais pas par 3.
3. Utilisez une boucle `while` pour trouver les 20 premiers nombres divisibles par 5, 7 et 11, et imprimez-les.
4. Écrivez un programme Python qui, à partir d'une liste `A`, imprime le plus petit nombre de cette liste.
5. Le plus petit nombre divisible par 2, 3 et 4 est 12. Trouvez le plus petit nombre divisible par tous les entiers compris entre 1 et 10.
6. Écrivez un programme Python qui, étant donné une liste `A`, trouve toutes les paires de deux nombres dans cette liste dont la somme est soit 14 soit 44.
7. Écrivez un programme Python qui, étant donné une liste `A`, trouve tous les triplets de trois nombres dans cette liste dont la somme est soit 14 soit 44.
8. Écrivez un programme Python qui imprime tous les triplets ordonnés d'entiers `(a,b,c)` avec $0 \leq a < b < c$ tels que $a + b + c = 15$.
9. Écrivez un programme qui demande à l'utilisateur de saisir deux entiers positifs, la hauteur et la longueur d'un parallélogramme et imprime un parallélogramme de cette taille avec des étoiles.

   Par exemple, si la hauteur était de 3 et la longueur de 6, ce qui suit serait imprimé :

   ```
   ******
    ******
     ******
   ```

   Si la hauteur était de 4 et la longueur de 2, ce qui suit serait imprimé :

   ```
   **
    **
     **
      **
   ```

10. Écrivez un programme Python qui demande à l'utilisateur un entier $n > 1$ et crée et imprime une liste contenant tous les diviseurs de n.
11. Écrivez un programme pour prendre un entier positif $n > 1$ de l'utilisateur et afficher si le nombre est un nombre *parfait*, un nombre *abondant* ou un nombre *déficient* :

    Nombre parfait. Un nombre parfait est un nombre où la somme de ses diviseurs propres (les nombres qui le divisent de manière égale sans lui-même) est égale au nombre.

    Nombre abondant. Un nombre abondant est un nombre où cette somme dépasse le nombre lui-même.

    Nombre déficient. Un nombre déficient est un nombre où cette somme est inférieure au nombre lui-même.

Exemple: 28 est parfait: $1+2+4+7+14=28$.
12 est abondant: $1+2+3+4+6=16>12$.
16 est déficitaire: $1+2+4+8=15<16$.

12. Écrivez une programme qui prend une liste `A` d'entiers (non trié), et calcule deux indices `i` et `j` tels que `A[j]-A[i]` soit maximisé.

    Vous devez écrire un algorithme qui utilise une seule boucle.

    `Example:` pour `A = [3,6,1,5,11,7,1]`, la solution est `j = 4, i = 6`.

13. Étant donnée une liste `A` de nombres (ils peuvent être positifs ou négatifs), trouvez les deux indices i et j tels que la somme des nombres compris entre i et j (inclus) en `A` soit maximum.

    Exemple: pour `A = [1,7,-3,6,-9,2]`, la réponse est $i=0$ et $j=3$ car $[1,7,-3,6]$ a la somme maximale.

14. (**Difficile**) On vous donne une liste `A` de $2n$ entiers, dont la somme vaut S. Vous jouez le jeu suivant contre un adversaire :

    Vous commencez. À votre tour, vous choisissez l'un des deux nombres situés aux extrémités actuelles de la liste (`A[0]` ou `A[2n-1]`). Vous ajoutez ce nombre à votre score et le retirez de la liste.

    Puis c'est au tour de votre adversaire, qui fait de meme: il choisit un nombre à l'une des deux extrémités restantes, l'ajoute à son score et le retire de la liste.

    Vous continuez ainsi alternativement jusqu'à ce que la liste soit vide.

    La personne dont les nombres récoltés totalisent au moins S/2 gagne!

    Montrez qu'il existe une algorithme pour vous, telle que quel que soit le jeu de l'adversaire, vous gagnez!