Session d'exercices – Les boucles imbriquées

Etoiles

Cet exercice présente les boucles imbriquées. Voici un exemple d'un tel assemblage :

```python
# Loop 1
for i in range(n): # Outer loop controls the rows
    # Loop 2
    for j in range(i): # Print leading spaces for alignment
        print(" ", end="")
    # Loop 3
    for k in range(n - i): # Print stars decreasing with each row
        print("*", end="")
    print() # Move to the next line after printing a row
```

Cette exemple a trois boucles (voir la Figure 1):

1. **Boucle 1 (Loop 1)**: Toutes les instructions de cette boucle sont exécutées n fois. La valeur de la variable n est fixée par l'utilisateur.
2. **Boucle 2 (Loop 2)**: Toutes les instructions de cette boucle sont exécutées i fois. La valeur de la variable i change avec chaque nouvelle itération de la Boucle 1.
3. **Boucle 3 (Loop 3)**: Toutes les instructions de cette boucle sont exécutées $n - i$ fois. La valeur de la variable i change avec chaque nouvelle itération de la Boucle 1.

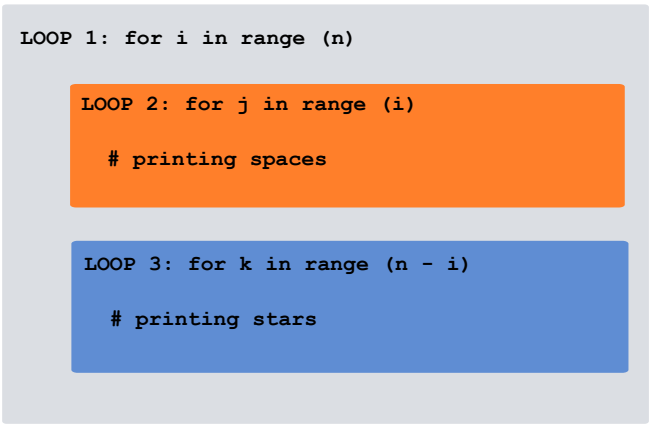


Figure 1: Les trois boucles imbriquées.

Imaginez maintenant que $n = 4$ et analysez comment le nombre d'itérations des boucles 2 et 3 change avec chaque itération de la boucle 1 (suivez les fleches et résultats partiels dans Figure 2):

- $i = 0$: la boucle 2 affiche aucun espace; ensuite, la boucle 3 affiche quatre etoiles;
- $i = 1$: la boucle 2 affiche un seul espace; ensuite, la boucle 3 affiche trois etoiles;
- $i = 2$: la boucle 2 affiche deux espaces; ensuite, la boucle 3 affiche deux etoiles;
- $i = 3$: la boucle 2 affiche trois espaces; ensuite, la boucle 3 affiche une etoile;
- la boucle 1 términe.

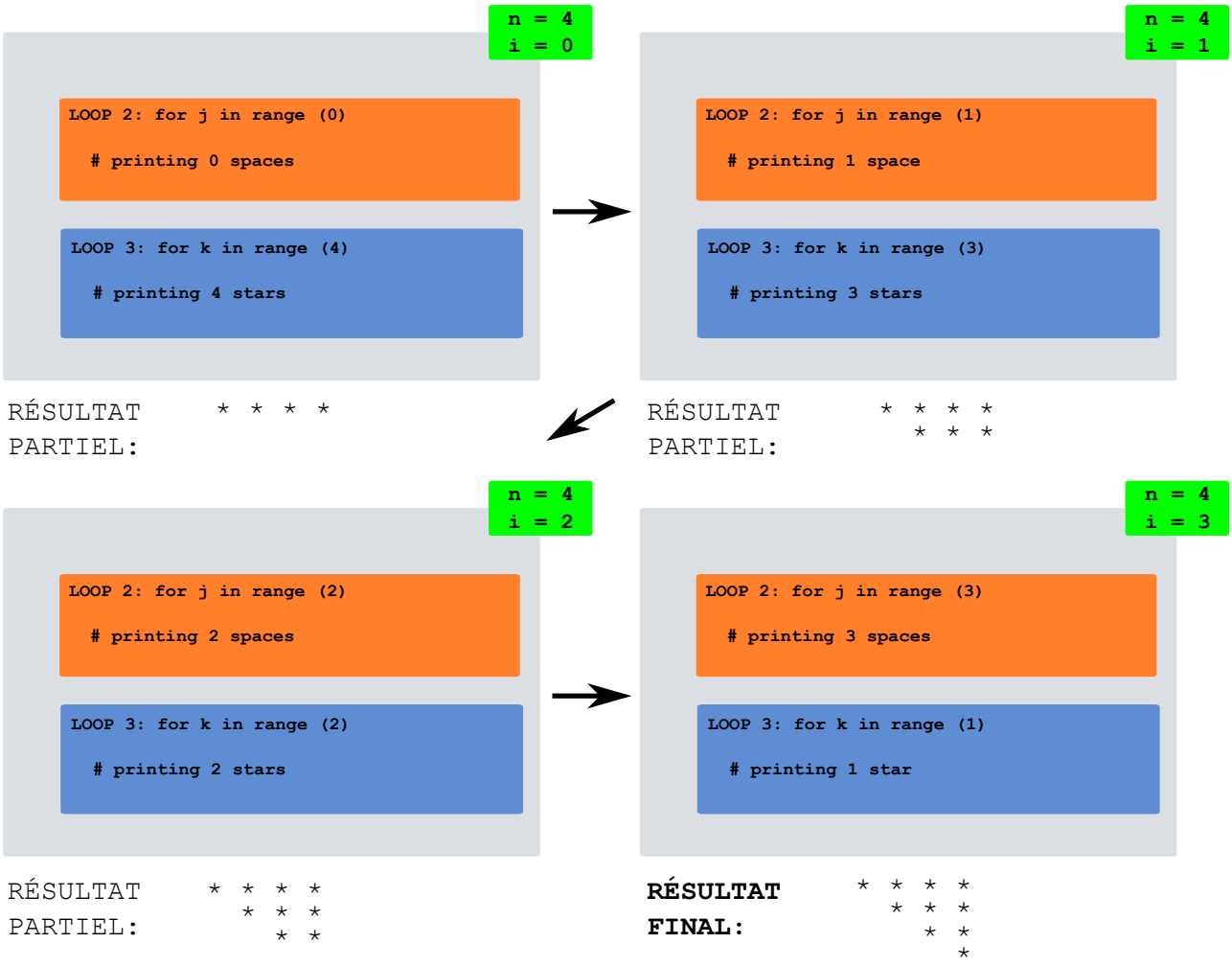


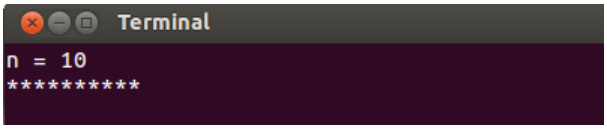
Figure 2: L'exécution du programme pour $n = 4$.

Dans les exercices suivantes, il faut créer un nouveau programme inspiré par ces trois boucles pour dessiner des formes différentes en utilisant des étoiles * et des espaces. Dans tous les programmes que vous écrirez, vous devez demander à l'utilisateur de fournir un nombre supérieur à zéro. Idéalement, vous pourriez faire en sorte que votre programme demande à plusieurs reprises jusqu'à ce que la valeur fournie soit conforme, mais cela n'est pas obligatoire. *Avant de coder, élaborez l'algorithme et testez-le sur papier pour vérifier qu'il fonctionne.*

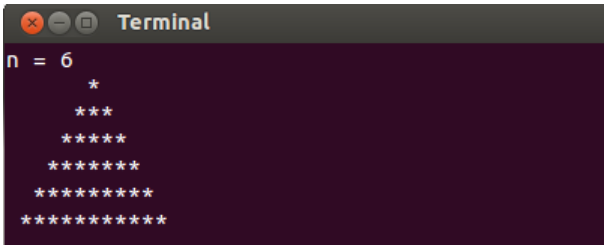
Dans l'exemple ci-dessous, nous fournissons le code complet correspondant à l'exemple précédent. Remarquez l'utilisation de `break` pour arrêter la boucle et de `continue` pour passer à l'itération suivante.

```python
while True: # Infinite loop to keep asking for valid input
    n = input("Enter a number larger than 0: ")
    if not n.isdigit(): # If the input is not a digit, repeat the loop
        continue

    n = int(n) # Convert input to an integer
    if n > 0: # Proceed only if the number is greater than 0
        for i in range(n): # Outer loop controls the rows
            for j in range(i): # Print leading spaces for alignment
                print(" ", end="")
            for k in range(n - i): # Print stars decreasing with each row
                print("*", end="")
            print() # Move to the next line after printing a row

        break # Exit the infinite loop once a valid input is received
```

1. [Difficulté: *] Créez un nouveau programme *stars_linear* pour dessiner n étoiles de façon linéaire.



2. [Difficulté: **] Créez un nouveau programme *stars_triangle* pour dessiner un triangle isocèle de hauteur n.



3. [Difficulté: ***] Créez un nouveau programme *stars_diamond* pour dessiner un losange avec le triangle isocèle du dessus de hauteur n.