

Exercice 1

```
# A l'aide d'une boucle, parcourir la liste suivante à la recherche de l'intrus
# Une fois l'intrus trouvé, printer "intrus trouvé!" et sortir de la boucle

liste = ["bernard", "gérard", "gontran", "jacqueline", "intrus", "nadia", "jack"]

for name in liste:
    if name == "intrus":
        print("intrus trouvé!")
        break
```

Exercice 2

```
# Sans utiliser la fonction sum(), retourner la somme de la liste suivante à
l'aide d'une boucle

liste_somme = [12.3, 34, 1, 0.4, 23, -17, 76, -300.2]

total = 0
for nombre in liste_somme:
    total += nombre

print(total)
```

Exercice 3

```
# A l'aide d'une boucle et d'une range, calculer le factoriel de 10 (3628800)
# Le factoriel d'un entier n est le produit des nombres entiers strictement
positifs inférieurs ou égaux à n
# Printer le résultat

total = 1
for x in range(1, 11):
    total *= x

print(total)
```

Exercice 4

```
# A l'aide d'une boucle while, demander à l'utilisateur de "Taper oui, ou non.",
et tant que ce dernier n'a pas tapé "non", continuer de lui demander "Taper oui,
ou non."
# Si l'utilisateur ne tape ni "oui", ni "non", continuer la boucle en lui mettant
un message d'erreur car l'input est invalide

question = input("Taper oui, ou non.")

while(question != "non"):
    question = input("Taper oui, ou non.")
    if question != "oui" and question != "non":
        print("Erreur dans la requête, vous devez taper 'oui' ou 'non' !")
```

Exercice 5

```
# A partir de la liste suivante printer le résultat suivant à l'aide d'une boucle
:
# "L'élément à l'index 0 est a"
# "L'élément à l'index 1 est 3"
# "L'élément à l'index 2 est True"
# ...

ma_liste = ['a', 3, True, "coucou", 'r', 3.14, [1, 2, 3]]

for index, elem in enumerate(ma_liste):
    print(f"L'élément à l'index {index} est {elem}")
```

Exercice 6

```
# A l'aide d'une compréhension de liste générer une nouvelle liste suivant les
règles suivante :
# Si le chiffre est un multiple de 5, le multiplier par 2
# Sinon, retourner sa division entière par 3
# Printer la nouvelle liste obtenue

liste_de_base = [23, 1, 27, 28, 3, 4, 763, 12, 90]
comp_liste = [x*2 if x%5 == 0 else x//3 for x in liste_de_base]

print(comp_liste)
```

Exercice 7

```
# A l'aide d'une compréhension de liste et de all() printer une fois True ou False
si toutes les chaînes de caractères contenues dans la liste sont des palindromes.

palindromes = ["kayak", "coloc", "malayalam", "pop", "erre"]
all_condition = all([mot[:] == mot[::-1] for mot in palindromes])

print(all_condition)
```

Exercice 8

A l'aide de boucles imbriquées, créer une nouvelle liste "flat", qui sera une liste aplatie de "liste", ayant les éléments classés dans l'ordre décroissant :
[7, 6, 5, 4, 3, 2, 1]

```
liste = [1, 3, 7, [4, 6, [5, 2]]]
flat = []
for root_element in liste:
    if isinstance(root_element, list):
        for elem in root_element:
            if isinstance(elem, list):
                flat.extend(elem)
            else:
                flat.append(elem)
    else:
        flat.append(root_element)

flat.sort(reverse=True)
print(flat)
```