Exercice 1

```
# Créer une fonction square_root prenant un nombre en paramètre, et retournant la racine carrée de ce nombre (vous pouvez utiliser le module math)
# Printer l'appel de la fonction avec un nombre donné
```

Exercice 2

```
# Créer une fonction extendator ne prenant aucun paramètre, et qui ne retourne rien.
# Faire en sorte qu'elle modifie la liste suivante définie dans l'espace global en l'étendant de la liste suivante [4, 5, 6]

liste_depart = [1, 2, 3]
```

Exercice 3

```
# Créer une fonction add prenant en paramètres deux nombres et qui retourne l'addition de ses deux nombres
# Créer une fonction carre qui prend un nombre en paramètre et retourne ce dernier élevé au carré
# A l'aide de ses deux fonction, retourner la somme des carrés de 2 et 3 (2² + 3²)
# Printer le résultat
```

OPTIONNEL : Exercice 4

```
# Le labyrinthe
# Partant de maze[0][0] (8) dans le labyrinthe, créer une fonction récursive capable de trouver la sortie (9)
# Les 0 présents dans le labyrinthe sont les murs infranchissables
# Les 1 sont des chemins possibles
# Parcours en diagonale interdit
# L'entrée du labyrinthe sera toujours à [0][0]
# La fonction fonctionne quelque soit la longueur et la largeur du labyrinthe
# La fonction ne prévoit pas le cas de figure suivant où quatre 1 sont côte à côte
```

```
:
# [[8, 1, 1*, 1*]
#  [0, 0, 1*, 1*] -> *quatre 1 en carré interdit boucle infinie probable
#  [0, 0, 0, 9]]

#    x = 0  1  2  3  4  5      y
maze = [[8, 1, 1, 1, 1, 0],  # 0
        [1, 0, 1, 0, 1, 0],  # 1
        [1, 0, 1, 1, 1, 1],  # 2
        [1, 0, 0, 0, 1, 0],  # 3
        [1, 1, 1, 0, 1, 0],  # 4
        [0, 1, 0, 0, 1, 9]]  # 5
```

Correction: 19. Les fonctions