

1. Code à comprendre :

Que donne l'exécution du script suivant :

```
1 def inutile(a) :
2     for i in range(10) :
3         mot = "DS"
4     return mot
5
6 b = inutile(2020)
7 print(b)
```

2. Code qui retourne le max de 2 nombres :

La fonction *maxi()* retourne le maximum des 2 nombres mis en argument. Compléter ci-dessous le code de cette fonction :

```
1 # Fonctions
2
3
4
5
6
7
8
9 # Programme principal
10 c = maxi(80, -5)
11 print(c)
12 print(maxi(-5, 80))
```

L'exécution de ce code donne :

```
>>> (executing file "exercice2.py")
80
80
```

3. Code qui échange 2 lettres d'une phrase :

La fonction *echange()* définit la page 2, prend en argument 3 chaînes de caractères : une phrase et 2 lettres. Elle retourne cette même phrase en ayant **échangé** les caractères identiques à la 1^{ère} lettre par la 2nd lettre :

Ainsi, en exécutant cette fonction comme cela dans le programme principal :

```
16 # Programme principal
17 first_phrase = echange_lettre("le coronavirus", "a", "e")
18 print(first_phrase)
19
20 next_phrase = echange_lettre("opéra", "a", "o")
21 print(next_phrase)
```

on obtient dans le shell :

```
>>> (executing file "exercice2.py")
la coronevirus
apéro
```

3.1. Compléter le script de la fonction *echange()* :

```
1 # Fonctions
2 def echange_lettre(phrase, lettre_1, lettre_2) :
3     phrase_modifiee = ""
4
5     for l in phrase :
6
7
8
9
10
11
12
13
14
15
```

3.2. Valeurs prises par les variables :

Pour la seconde exécution : `next_phrase = echange_lettre("opéra", "a", "o")`

Compléter le tableau ci-dessous qui donne les valeurs des différentes variables utilisées, au cours de cette exécution.

<i>phrase</i>	<i>lettre_1</i>	<i>lettre_2</i>	<i>ℓ</i>	<i>phrase_modifiee</i>	<i>next_phrase</i>

4. Fonction qui compare 2 nombres :

```
# Fonctions
```

```
# Programme principal
c = compare(80, -5)
print(c)
print(compare(-5, 80))
print(compare(-5, -5))
```

L'exécution du script précédent donne :

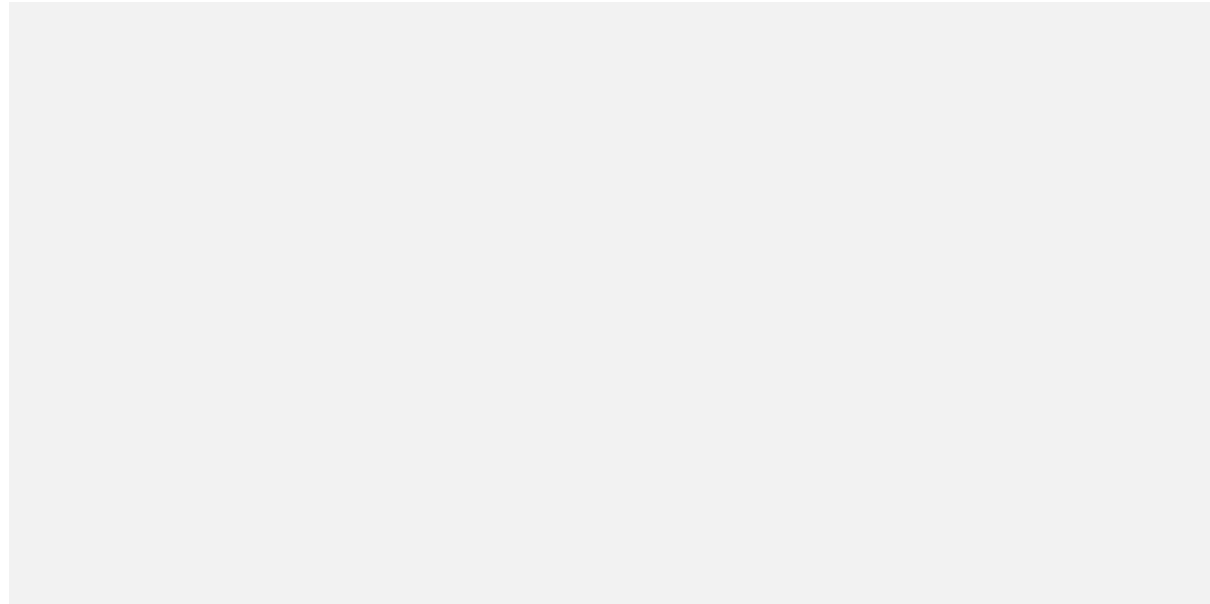
⇒ Compléter ce script.

(< , = , > , sont des caractères comme les autres)

```
>>> (executing file "exercice2.py")
80 > -5
-5 < 80
-5 = -5
```

5. Code qui donne des punitions au lycée :

```
# fonctions
```



```
# main
message1 = punition("bavardages","non")
print(message1)

message2 = punition("bavardages","oui")
print(message2)

message3 = punition("insultes","non")
print(message3)

message4 = punition("insultes","oui")
print(message4)

message5 = punition("portable","non")
print(message5)

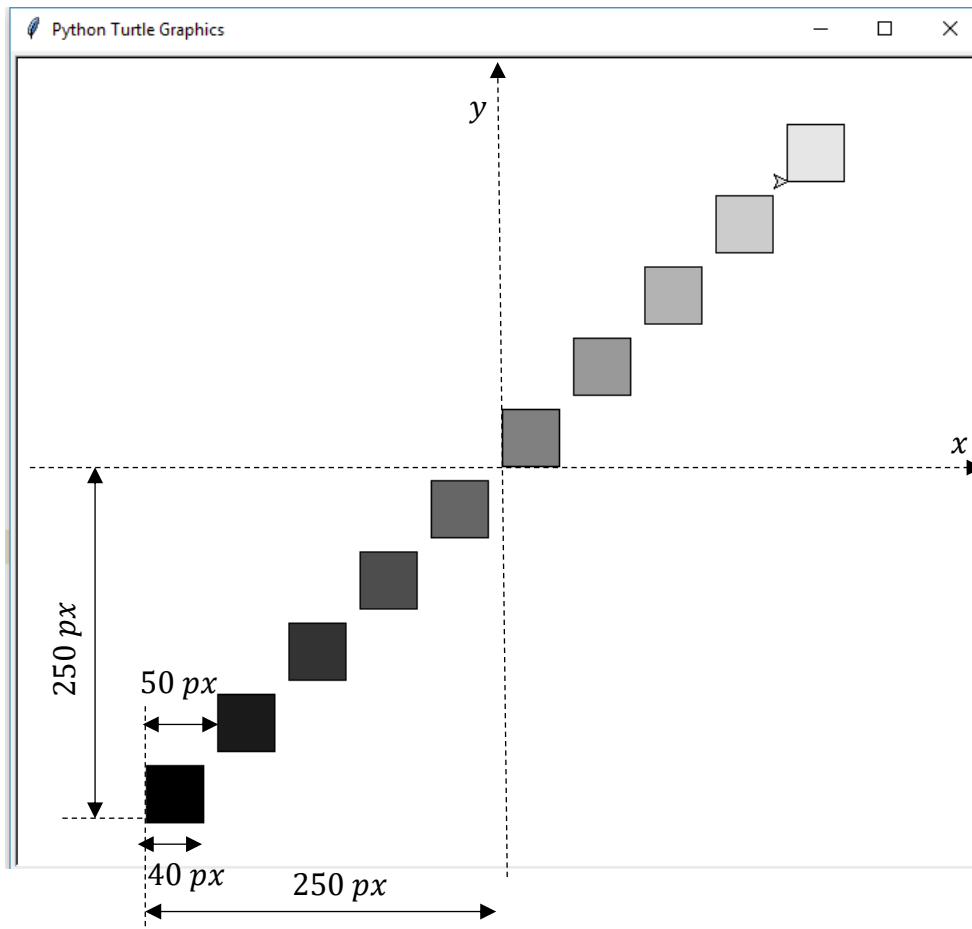
message6 = punition("portable","oui")
print(message6)
```

Le code précédent permet d'obtenir dans le shell cela :

⇒ Compléter ce code.

```
>>> (executing file "exercice5.py")
bavardages : 2 heures de colles
bavardages et récidive : 4 heures de colles
insultes : 3 heures de colles
insultes et récidive : 6 heures de colles
portable : 4 heures de colles
portable et récidive : 8 heures de colles
```

6. Code qui trace des carrés :



En exécutant le code qui suit et qui est incomplet, on obtient la fenêtre graphique ci-dessus.

⇒ Compléter ce code en utilisant une boucle `for ... in range()` si possible.

```
from turtle import *

# Fonctions
def carre(a,x,y,r,g,b) :
    up()
    goto(x,y)
    fillcolor(r,g,b)
    down()
    begin_fill()
    for i in range(4) :
        forward(a)
        left(90)
    end_fill()

# Main

exitonclick()
```