

Exercice 1 :

⇒ Donner un script possible pour la fonction *boite* :

```

1 # fonctions
2 def boite
3
4
5
6 # programme principal
7 assert boite(1) == 0
8 assert boite(2) == 2
9 assert boite(3) == 0
10 assert boite(4) == 4
11 assert boite(5) == 0
12 assert boite(6) == 6

```

Exercice 2 : ⇒ Remplir le tableau ci-dessous, donnant la valeur des différentes variables du script ci-contre, durant son exécution :

<i>n_bin</i>	<i>i</i>	<i>n_dec</i>	<i>c</i>

```

1 # Définition des fonctions *****
2 def nbr_caractere(chaine) :
3     """
4     Chaîne est une chaîne de caractère
5     Retourne le nombre de caractère de chaîne
6     """
7     n = 0
8     for c in chaine :
9         n = n + 1
10    return n
11
12
13 def conversion_binaire(n_bin):
14     """
15     n_bin est un nombre en base 2, composé de 0 et de 1
16     Retourne la valeur en base 10 de n_bin
17     """
18     n_bin = str(n_bin)
19     i = nbr_caractere(n_bin) - 1
20     n_dec = 0
21     for c in n_bin :
22         n_dec = n_dec + int(c)*2**i
23         i = i - 1
24     return n_dec
25
26 # Main *****
27 n_dec = conversion_binaire(101)
28 print(n_dec)

```

```

1 def trouve
2
3
4
5
6
7
8
9 assert trouve("Bonjour","o") == 2
10 assert trouve("Bonjour","r") == 1
11 assert trouve("nsi","r") == 0

```

Exercice 3 :

⇒ Donner un script possible pour la fonction *trouve* :

Exercice 4 : ⇒ Remplir à nouveau le tableau donnant la valeur des différentes variables du script ci-contre, pour cette nouvelle exécution

<i>n_bin</i>	<i>i</i>	<i>n_dec</i>	<i>c</i>

```

13 def conversion_binaire(n_bin):
14     """
15     n_bin est un nombre en base 2, composé de 0 et de 1
16     Retourne la valeur en base 10 de n_bin
17     """
18     n_bin = str(n_bin)
19     i = nbr_caractere(n_bin) - 1
20     n_dec = 0
21     for c in n_bin :
22         n_dec = n_dec + int(c)*2**i
23         i = i - 1
24     return n_dec
25
26 # Main *****
27 n_dec = conversion_binaire(1100)
28 print(n_dec)

```

Exercice 5 : On donne le script de la fonction ci-contre :

⇒ Remplir le tableau donnant la valeur des différentes variables du script ci-contre, pour les 2 exécutions ci-dessous :

```

1 def affectation(n_hex) :
2     n = 0
3     for i in "0123456789abcdef" :
4         if n_hex == i :
5             return n
6         else :
7             n = n + 1
8     return n

```

10 n_dec = affectation("4")

<i>n_hex</i>	<i>n</i>	<i>i</i>	<i>n_dec</i>

10 n_dec = affectation("a")

<i>n_hex</i>	<i>n</i>	<i>i</i>	<i>n_dec</i>

Exercice 6 : On donne le script de la fonction ci-contre :

⇒ Remplir le tableau donnant la valeur des différentes variables du script ci-contre, lors de son exécution :

```

1 def coupe(mot, longueur) :
2     n = 1
3     gauche = ""
4     droite = ""
5     for c in mot :
6         if n <= longueur :
7             gauche = gauche + c
8         else :
9             droite = droite + c
10            n = n + 1
11    return gauche , droite
12
13 m , n = coupe("Bonjour", 2)
14 print(m)
15 print(n)

```

<i>mot</i>	<i>longueur</i>	<i>n</i>	<i>gauche</i>	<i>droite</i>	<i>c</i>	<i>m</i>	<i>n</i>

Exercice 7 :

⇒ Donner un script possible pour la fonction *hache* qui retourne 1 caractère sur 2 du String en argument :

⇒ Construire un tableau donnant l'état des variables durant l'exécution.

```

1 def
2
3
4
5
6
7
8
9
10 assert hache("Sapin_de_noel") == "ai_ene"

```

Exercice 8 :

⇒ Remplir le tableau donnant la valeur des différentes variables du script ci-contre, lors de son exécution :

```

1 # Fonctions
2 def echange_lettre(phrase,lettre_1,lettre_2) :
3     phrase_modifiee = ""
4
5     for l in phrase :
6         if l == lettre_1 :
7             phrase_modifiee = phrase_modifiee + lettre_2
8         elif l == lettre_2 :
9             phrase_modifiee = phrase_modifiee + lettre_1
10        else :
11            phrase_modifiee = phrase_modifiee + l
12
13    return phrase_modifiee
14
15
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)
22

```

phrase	lettre_1	lettre_2	phrase_modifiée	ℓ	first_phrase

phrase	lettre_1	lettre_2	phrase_modifiée	ℓ	first_phrase