

Exercices d'algorithmique n°1

EXERCICE 1 : Écrire une fonction `double` qui prend un nombre `x` et qui renvoie le double du nombre `x`

```
def double(x):  
    return 2*x
```

EXERCICE 2 : Écrire une fonction `carre` qui prend un nombre `x` et qui renvoie le carré du nombre `x`

```
def carre(x):  
    return x * x
```

EXERCICE 3 : Écrire une fonction `prod_pos` qui prend des nombres `x` et `y`, et qui renvoie un booléen qui indique si le produit de `x` par `y` est supérieur ou égal à 0.

```
def prod_pos(x, y):  
    return x*y >= 0
```

EXERCICE 4 : Écrire une fonction `mention` qui prend un nombre `note` et qui affiche Echec, Pas de mention ou Mention, si `note < 10`, $10 \leq \text{note} < 12$ ou `note \geq 12`.

```
def mention(note):  
    if note >= 12:  
        print("Mention")  
    elif note >= 10:  
        print("Pas de mention")  
    else:  
        print("Echec")
```

EXERCICE 5 : Écrire une fonction `somme_carres` qui prend un entier `n` et qui renvoie la somme des carrés de 1 à `n` inclus. Par exemple : `somme_carres(3) = 12 + 22 + 32 = 14`.

```
def somme_carres(n):  
    somme = 0  
    for i in range(1, n+1): # Pour aller de 1 à n inclus  
        somme = somme + i*i  
    return somme
```

EXERCICE 6 : Écrire une fonction `longueur` qui prend une chaîne de caractères `texte` et qui renvoie un entier correspondant au nombre de symboles de `texte`. On n'utilisera pas la fonction `len`.

```
def longueur(texte):  
    c = 0  
    for s in texte:  
        c = c + 1  
    return c
```

EXERCICE 7 : Écrire une fonction `appartient` qui prend une chaîne de caractères `texte` et un symbole `symbole`, et qui renvoie un booléen indiquant si `symbole` est contenu dans `texte`.

```
def appartient(symbole, texte):  
    for s in texte:  
        if s == symbole:  
            return True  
    return False
```

EXERCICE 8 : Écrire une fonction `compter` qui prend une chaîne de caractères `texte` et un symbole `symbole`, et qui renvoie un entier correspondant au nombre d'occurrences de `symbole` dans `texte`. On n'utilisera pas la fonction `count`.

```
def compter(symbole, texte):  
    c = 0  
    for s in texte:  
        if s == symbole:  
            c = c + 1  
    return c
```

EXERCICE 9 : Écrire une fonction `indice` qui prend une chaîne de caractères `texte` et un symbole `symbole`, et qui renvoie un entier correspondant à l'indice de la première occurrence de `symbole` dans `texte`. S'il n'y en a pas, la fonction renvoie `-1`. On n'utilisera pas `index`.

```
def indice(symbole, texte):  
    pos = 0  
    for s in texte:  
        if s == symbole:  
            return pos  
        pos = pos + 1  
    return -1
```

EXERCICE 10 : Écrire une fonction `nombres_de_multiples` qui prend 3 entiers `k1`, `k2` et `n`, et qui renvoie le nombre d'entiers qui sont multiples de `k1` ou de `k2` et qui sont strictement inférieurs à `n`. On rappelle que `m` est un multiple de `d` si `m%d` vaut `0`.

```
def nombres_de_multiples(k1, k2, n):  
    compteur = 1  
    for i in range(n):  
        if i % k1 == 0 and i % k2 == 0:  
            compteur = compteur + 1  
    return compteur
```