

Exercices d'algorithmique n°1

EXERCICE 1 : Écrire une fonction `double(x)` qui renvoie le double du nombre `x`

```
def double(x):
```

EXERCICE 2 : Écrire une fonction `carre(x)` qui renvoie le carre du nombre `x`

```
def carre(x):
```

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

```
def prod_pos(x, y):
```

EXERCICE 4 : Écrire une fonction `mention(note)` qui affiche Echec, Pas de mention ou Mention, si $note < 10$, $10 \leq note < 12$ ou $note \geq 12$.

```
def mention(note):
```

EXERCICE 5 : Écrire une fonction `somme_carres(n)` qui renvoie la somme des carrés de 1 à `n` inclus. Par exemple : $somme_carres(3) = 1^2 + 2^2 + 3^2 = 14$.

```
def somme_carres(n):
```

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

```
def longueur(texte):
```

EXERCICE 7 : Écrire une fonction `appartient(symbole, texte)` qui renvoie un booléen indiquant si le symbole `symbole` est contenu dans `texte`. On n'utilisera pas la commande `in`.

```
def appartient(symbole, texte):
```

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

```
def compter(symbole, texte):
```

EXERCICE 9 : Écrire une fonction `position(symbole, texte)` qui renvoie un entier correspondant à la position de la première occurrence de `symbole` dans `texte`. S'il n'y est pas, la fonction renvoie `-1`. On n'utilisera pas la fonction `index`.

```
def position(symbole, texte):
```

EXERCICE 10 : Écrire une fonction `nombres_de_multiples(k1, k2, n)` qui renvoie le nombre de nombres 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):
```