

I. D erouler des algorithmes

Pour chacun des exemples de codes Python suivants, indiquer ce que vaut la variable x en fin de programme.

```
# Exemple 1
a = 3
b = 4
c = 5
x = 0
if a * c < b**2:
    x += 1
elif a + x > b:
    x += 2
    b += 2
if b < c:
    x += 4
```

```
# Exemple 2
n = 535
x = 0
while n > 0:
    x += n % 10
    n //= 10
```

```
# Exemple 3
e = 1
x = 0
for i in range(5):
    x += e * i
    e *= 2
```

II. Relire un programme : Le jeu des 7 erreurs

Le programme suivant a pour but de calculer la somme des entiers de 0   100 (inclus) qui sont multiples de 2 ou 3. En l' tat, ce code ne tourne pas car 7 erreurs s'y sont gliss es. Identifier-les et corriger-les.

```
k = 0
while k < 100
    if k % 2 = 0:
        s += k
    else k % 3 = 0:
        s += k
print(s)
```

III. Comprendre un algorithme

On consid re la fonction `f` d finie par le code Python suivant :

```
def f(n):
    e = 1
    s = 0
    for i range(n):
        s += e
        e *= -1
    return s
```

4. Que retourne la fonction si on calcule $f(2)$, $f(3)$, $f(4)$, $f(5)$, $f(6)$?
5. Pouvez-vous résumer en quelques mots simples ce que fait la fonction f ?
6. Pouvez-vous prévoir ce que retournerait le calcul de $f(100000)$?
7. Répondre aux mêmes questions avec la fonction f suivante :

```
def f(p):  
    d = 2  
    while d**2 <= p and p % d != 0:  
        d += 1  
    if p % d == 0:  
        return False  
    else :  
        return True
```

IV. Ecrire un algorithme

Ecrire un algorithme (en pseudo-code ou en Python) qui compte le nombre de « smileys » tapés au clavier par un utilisateur. Ce dernier a terminé la saisie lorsqu'il a tapé la caractère # (dièse).

On se limitera aux 3 smileys ci-dessous, codifiés par 3 caractères :

- le smiley → :-)
- le sad → :-(
- wink → ;-)

Par exemple, si l'utilisateur tape la chaîne suivante :

```
"Je te remercie pour l'invitation :-), mais je ne pourrai pas  
venir :-(  
Je ;-( propose de reporter à une prochaine fois ;-)"
```

L'algorithme doit afficher :

```
Vous avez utilisé 3 smileys !
```