

Soigner sa copie!

A Pourquoi?

Il est nécessaire d'apporter un soin particulier à la copie lors des examens et des concours. Comme vous pourrez le constater, ce n'est pas toujours le cas. **Si votre copie n'est pas claire, lisible et intelligible, cela ne sera jamais à votre avantage.**

B Extrait de copies de concours

Ci-dessous, des extraits de copie qui illustrent les choses à faire et à ne pas faire.

A) `def monte_carlo(s, T, m_tests):`
`for k in range(m_tests):`
 `e = s[0]`
 `e = randrange(n)`
 `delta_e = calcul_delta_e2(s, e)`
 `if test_boltzmann(delta_e, T):`
 `s[e] = s[e] * (-1)`
`return s`

FIGURE 1 – Copie non commentée, bien indentée et intelligible. La variable n est une variable globale.

16) `def monte_carlo(s, T, m_tests):`
`for i in range(m_tests):` # on le fait m_tests fois
 `j = randrange(n)` # doigt de l'indice au hasard
 `delta = calcul_delta_e2(s, j)` # calcul du DE
 `if test_boltzmann(delta, T) == True:` # si il faut
 `s[i] = -s[i]` # on change le signe du spin
`return s`

FIGURE 2 – Copie commentée mais les commentaires n’apportent rien et ce n’est pas plus intelligible. L’initialisation du j à zéro est inutile car la variable est locale à la boucle et initialisée par la fonction randrange. Le test est tautologique (==True inutile), on ne dit pas «si il» mais «s’il». Enfin, c’est s[j] pas s[i] d’après les instructions du sujet.

```

def monte_carlo(s, T, m_tests):
    for k in range(m_tests):
        p = math.random(len(s))
        if test_boltzmann(celul_delta_e2(s, p), T):
            s[p] = -1 * s[p]
    return s
    # 6. fait m_tests tests
    # O prend en indice au hasard
    # Si le spin change, alors
    # on change la ligne en
    # le multplié par (-1).

```

FIGURE 3 – Copie commentée mais les commentaires n'apportent rien et en plus c'est **illisible**.

```

def dichot(f, t, a, b, eps):
    while (b-a) > eps:
        m = (a+b)/2
        if f(b,t)*f(m,t) > 0:
            b = m
        else:
            a = m
    return (a+b)/2

```

FIGURE 4 – Copie très lisible et intelligible, avec marquage de l'indentation

```

3) def dichot(f,t,a,b,eps):
    m=(a+b)/2 # on s'intéresse au milieu de l'intervalle [a,b]
    while b-a > 2*eps: # si cette condition n'est pas satisfait, c'est
        que le point qu'on a est à moins d'eps du zéro de la fonction
        if f(m)*f(a)>0: # le zéro n'est pas dans l'intervalle [a,m]
            a=m # on déplace la bordure gauche de l'intervalle
        else:
            b=m # on déplace la bordure droite
        m=(a+b)/2 # on prend le milieu du nouvel intervalle
    return m

```

FIGURE 5 – Copie commentée mais pas plus intelligible. **Les commentaires rendent l'indentation illisible. Le return sur une nouvelle page est très maladroit (le correcteur pourrait le rater...).** En plus c'est faux, c'est $>\text{eps}$ et pas $>2*\text{eps}$.

```

def dichoto(f, t, a, b, eps):
    while f(b-a) >= eps:
        m = (a+b)/2
        if f(a, t) * f(m, t) > 0: # 2eps, car on
            # prend la moitié
            # à la fin
            a = m
            # où se trouve
            # le zero
        elif f(a, t) * f(m, t) < 0:
            b = m
        else:
            return m
            # est tel que f(m)
            # = 0
    return (a+b)/2

```

FIGURE 6 – Copie commentée mais pas plus intelligible. L'indentation est ratée et guère lisible. Il y a confusion entre l'opérateur de test == et l'opérateur d'affectation =. Enfin, c'est >eps et pas >2*eps et on aurait pu faire avec moins de cas.

```

7. SELECT nom-fournisseur, prix-hg * 4.5
   FROM fournisseurs JOIN prix
   ON id-fournisseur = id-far
   WHERE id-mot = 8713

```

FIGURE 7 – Copie intelligible

7) SELECT nom-fournisseur , 4,5 * prix/kg
FROM fournisseurs
JOIN prix ON id-fournisseur = id-fourn
WHERE id-mat = 8713

FIGURE 8 – Copie mal écrite mais qui reste intelligible. Comme quoi quand on veut on peut!