

1

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** pour plusieurs raisons :

- un code mal présenté est souvent défaillant,
- un raisonnement mal exprimé est la plupart du temps faux,
- cela énerve le correcteur qui sera très prompt à ne rien laisser passer en prétextant l'illisibilité,
- un devoir ou une copie d'examen c'est une forme de relation épistolaire : vous parlez à celui qui corrige la copie. La moindre des politesses, c'est de s'exprimer clairement, lisiblement et intelligiblement.

B Conseils pour la rédaction

- De la clarté avant toute chose, de l'aération dans la présentation :
 1. bien séparer les questions,
 2. ne pas hésiter à sauter des lignes pour aérer la réponse,
 3. **procéder dans l'ordre des questions**, laisser de la place si nécessaire,
 4. commencer une nouvelle partie sur une nouvelle copie,
 5. laisser de la place (suffisamment) lorsque vous n'abordez pas certaines questions.
Vous pourrez toujours tenter d'y répondre plus tard.
- Faire des schémas explicites pour expliquer des stratégies,
- Mettre en évidence la réponse finale à la question (par exemple une complexité),
- Ne pas utiliser de correcteurs, d'effaceur ou de stylo à friction, car cela pose des problèmes à la numérisation des copies : rayer proprement, sauter une ligne et recommencer correctement,
- Faire un effort si votre écriture n'est pas très lisible.

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

FIGURE 1.1 – Copie non commentée, bien indentée et intelligible. La variable n est une variable globale.

C Extraits de copies de concours illustrant ce qu'il faut faire ou ne pas faire

C. EXTRAITS DE COPIES DE CONCOURS ILLUSTRANT CE QU'IL FAUT FAIRE OU NE PAS FAIRE

```

16) def monte_carlo(s, T, m_tests):
    for i in range(m_tests):          # on le fait m_tests fois
        j = randrange(m)              # choix 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 1.2 – 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 ($==\text{True}$ 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):          # k fait m_tests tests
        p = math.randrange(len(s))    # p prend un indice au hasard
        if test_boltzmann(calcul_delta_e2(s, p), T):
            s[p] = -1 * s[p]           # si le spin change, alors
                                         # on change sa ligne en
                                         # le multiplié par (-1).
    return s

```

FIGURE 1.3 – 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 1.4 – Copie très lisible et intelligible, avec marquage de l'indentation

C. EXTRAITS DE COPIES DE CONCOURS ILLUSTRANT CE QU'IL FAUT FAIRE OU NE PAS FAIRE

```
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,b) > 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 1.5 – 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 cherche(f, t, a, b, eps):
    while ((b-a)>2*eps):
        m = (a+b)/2
        if f(a,t)*f(m,t)>0: # où se trouve
            | a=m                                # le zero
        elif f(a,t)*f(m,t)<0:
            | b=m
        elif f(m,t)=0: # cas où a=b
            | return m                          # est tel que f(m)
        else:                                     # f(m)≠0
            | return a
    return (a+b)/2

```

FIGURE 1.6 – 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 1.7 – Copie intelligible

C. EXTRAITS DE COPIES DE CONCOURS ILLUSTRANT CE QU'IL FAUT FAIRE OU NE PAS FAIRE7

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

FIGURE 1.8 – Copie mal écrite mais qui reste intelligible. Comme quoi quand on veut on peut!