

Conseils pour les épreuves écrites

Conseils pratiques pour la bonne (à mon sens, qui n'engage que moi et ma modeste expérience de correcteur/examineur) rédaction des Mathématiques et de l'Informatique en concours.

1. Concernant les Mathématiques

Le début de votre copie est important : si elle présente des points positifs, vous mettrez le correcteur en confiance pour le reste de l'épreuve et serez évalué avec *bienveillance*. Attention, je ne dis pas ici que l'on peut faire n'importe quoi dans la suite mais il faut donc entre autres éviter impérativement en début de copie :

- ✓ les tentatives d'escroquerie,
- ✓ les ratures et tâches d'effaceur,
- ✓ les calculs inachevés,
- ✓

1— 1. Le brouillon

Le brouillon sert à chercher des solutions, à effectuer des calculs délicats, ce qui permet d'éviter les ratures. Rédiger un devoir ne consiste pas à recopier un brouillon, mais à le repenser pour obtenir la solution la plus brève possible.

1— 2. La présentation

- ✓ Les textes des questions ne sont pas recopiés mais les numéros oui et avec un fort contraste avec le reste ; cela évitera de perdre le correcteur et ainsi d'éviter tout oubli de points si le correcteur ne voit pas le début d'une nouvelle question. Quelque chose du type **2.1** est beaucoup plus voyant qu'un « 2.1 » perdu en milieu de texte. Utilisez des feutres pour clairement les faire apparaître.
- ✓ Les résultats doivent être **encadrés** et mis en évidence.
- ✓ L'articulation des démonstrations (donc, or, si, alors, ...) doit être mise en évidence elle aussi, les hypothèses, les conclusions et surtout le type de démonstration (récurrence, absurde, contraposée, analyse/synthèse, double inclusion pour l'égalité de deux ensembles, etc.).
- ✓ L'usage des produits blancs et effaceurs doit rester exceptionnel.

1— 3. La rédaction

- ✓ Les phrases seront courtes, grammaticalement correctes, et dans la mesure du possible sans fautes d'orthographe.
- ✓ On n'emploiera pas les symboles mathématiques comme des abréviations et on les évitera en général ; notamment pas de « hyp », « ccl » etc. .
- ✓ Attention aux non-sens Mathématiques ; en 2^{ème} année de CPGE, les phrases du type « la fonction $f(x)$ est » ne sont plus autorisées !
- ✓ On ne mélangera pas les quantificateurs avec le reste du texte. Notamment « on voit que $\forall x$, la fonction f est ». Une rédaction correcte est par exemple : « la fonction f vérifie : $\forall x \in \dots, \dots$ ».
- ✓ Les formules du genre « on voit bien que », « il est évident que », etc. sont à proscrire ; ceci démontre généralement l'incapacité du candidat à effectuer ladite preuve.
- ✓ S'interroger : se poser les questions suivantes. Les calculs ne sont-ils pas exagérément longs ? la méthode utilisée est-elle judicieuse ? les résultats sont-ils cohérents ? cohérents avec le texte ? Ne pas hésiter à ajouter des remarques si vous détectez des incohérences, votre regard critique sur ce que vous faites sera apprécié.
- ✓ L'utilisation excessive des symboles \iff , \implies ou \implies est à éviter. Cela vous évitera d'affirmer des équivalences lorsqu'elles ne sont pas vraies (et bien souvent on ne sert pas du sens indirecte dans des calculs littéraux).
- ✓ En revanche, lorsqu'il s'agit de caractériser les éléments d'un ensemble (exercice type : équation différentielle, ensemble de fonctions vérifiant une certaine propriété, etc.), il faut impérativement y avoir recours et préciser les sens non évidents (et le vérifier le cas échéant).

2. Concernant l'Informatique

Il faut respecter, pour plus de clarté, les règles que vous suivez sur n'importe quel IDE Python qui font partie intégrante du langage. En particulier :

- ✓ On indente toujours ses programmes, et aux endroits appropriés. Pour plus de clarté, utiliser un trait vertical sur la gauche pour préciser l'indentation.
- ✓ On commente (lorsque le correcteur voit 400 programmes sur 400 copies différentes, plus vous serez clairs, plus vous aurez de chance d'avoir les points).
- ✓ Généralement les programmes à créer sont courts. Il est déraisonnable d'imaginer qu'un programme d'une page soit nécessaire pour traiter une question. Allez au plus simple et au plus rapide!
- ✓ En parlant de simplicité, pensez à des versions récursives de vos programmes. Lorsque cela est possible, ce sera très apprécié!
- ✓ On n'oublie pas d'importer les bibliothèques nécessaires au bon déroulement de vos fonctions.