

PROPOS INTRODUCTIFS

Le sujet de l'oral d'informatique CCINP en spécialité MPI est constitué de deux exercices :

- un exercice de type A "théorique", noté sur 8 points et ne nécessitant pas l'utilisation d'un ordinateur. L'exercice A peut évaluer la syntaxe SQL mais ni la syntaxe C ni la syntaxe OCaml ;
- un exercice de type B "pratique", noté sur 12 points et consacré principalement à de la programmation en C ou en OCaml. Quelques questions d'analyse peuvent également être posées. Cet exercice fournit généralement un code compagnon que le candidat devra compléter, corriger ou modifier. Il est également possible que le candidat doive écrire intégralement le code demandé par l'énoncé.

L'oral, d'une durée totale d'une heure, se décompose en deux temps :

- 30 minutes de préparation du sujet sur feuille et machine, démarches administratives incluses ;
- 30 minutes de passage avec un examinateur, démarches administratives incluses.

Ce rapport rappelle les consignes principales, qui sont susceptibles d'évoluer. Se référer à la notice du Concours pour obtenir l'ensemble de l'information et au site du concours <https://www.concours-commun-inp.fr/fr/epreuves/les-epreuves-orales.html> rubrique MPI. Vous trouverez le cadre de l'épreuve (dont l'environnement technique sous format fichier .ova).

Également, restent disponibles les 12 « exercices 0 » qui furent proposés à la rentrée 2022-23, en amont de la première session Orléans 2023 de la filière MPI, afin que les candidats se préparent au mieux à cette nouvelle épreuve.

En complément à ce rapport, une sélection de sujets, commentés et partiellement corrigés, posés lors de la session 2024 est publiée, ainsi que l'archive correspondante contenant les codes compagnons, les codes corrigés et le fichier source LaTeX.

PRÉPARATION DU SUJET

La préparation s'effectue dans une salle dédiée. Plusieurs candidats préparent en même temps, dans des conditions favorisant la concentration, sous la surveillance de vacataires. À l'entrée de la salle de préparation, convocations et pièces d'identité sont vérifiées.

Pour préparer, le candidat dispose :

- d'un ordinateur disposant de l'environnement diffusé sur le site du concours à la rubrique MPI¹ et d'un clavier (configuration AZERTY par défaut),
- de feuilles de brouillon,
- des énoncés imprimés des deux exercices (types A et B),

¹ <https://www.concours-commun-inp.fr/fr/epreuves/les-epreuves-orales.html>

- d'une calculatrice,
- d'une clé USB contenant le code accompagnant l'exercice de type B.

Le candidat doit copier le code compagnon disponible sur la clé sur sa machine pour pouvoir le compléter / modifier, puis copier le code produit dans un répertoire dédié sur sa clé USB en fin de préparation. Les consignes quant aux modalités de préparation et notamment la manipulation des clés sont projetées dans la salle de préparation, de même qu'un chronomètre indiquant le temps de préparation restant. Ce dernier est fixé à 27 minutes, les 3 minutes restantes étant réservées aux formalités administratives et au déplacement vers les salles d'interrogation.

En fin de préparation, les candidats sont accompagnés jusqu'aux salles d'interrogation par un vacataire.

PASSAGE DU CANDIDAT

Le candidat rentre dans la salle qui lui est indiquée, fournit à l'examineur sa feuille de passage et sa pièce d'identité puis signe la feuille d'émargement que lui présente l'examineur. Le chronomètre est lancé à ce moment, pour un temps de passage effectif de 27 minutes.

Le candidat dispose d'un ordinateur prêt, dans la même configuration que celui qu'il a utilisé dans la salle de préparation. Un vidéoprojecteur permet de présenter son travail à l'examineur. Le candidat est invité à copier le code présent sur sa clé sur la machine mise à disposition.

Le candidat commence par présenter tout ce qu'il a préparé. Il débute par l'exercice de son choix et peut, à tout moment, changer d'exercice et/ou revenir sur l'exercice précédemment abordé. Il dispose pour sa présentation d'un tableau et de son code source projeté. L'examineur demandera que le code produit soit compilé / exécuté / interprété. L'examineur peut donner des indications quant au temps restant et, le cas échéant, indiquer au candidat sur quelles questions se concentrer.

En fin d'épreuve, après la fin du temps imparti, le candidat remet à l'examineur les énoncés des deux exercices, ses brouillons et la clé contenant le code produit. Il ferme sur l'ordinateur les fenêtres de son code, efface le tableau avant de récupérer sa pièce d'identité et sa feuille de passage signée par l'examineur.

CRITÈRES D'ÉVALUATION

Sont pris en compte dans l'évaluation :

- la maîtrise des éléments du programme relatifs aux deux exercices,
- la maîtrise de l'environnement technique et des outils mis à disposition,
- la pertinence de la réflexion,
- l'interactivité lors de la présentation des résultats et des tests (exercice de type B),
- la réactivité aux éventuels conseils et indications de l'examineur,
- la qualité de la prestation orale.

BILAN

Sur 684 candidats admissibles (dont 104 grands admissibles), 526 se sont présentés aux épreuves orales. La moyenne des notes obtenues est de 11,50 avec un écart-type de 3,48.

Les examinateurs ont proposé des exercices couvrant l'ensemble du programme des deux années de MPI. En particulier, les thématiques suivantes ont été abordées : la récursivité, la représentation de types, les structures de données (listes, tableaux, piles, files, tas, etc.), les arbres, les graphes, les algorithmes d'approximation, les algorithmes probabilistes, les stratégies algorithmiques (diviser pour

régner, programmation dynamique, retour sur trace, etc.), l'algorithmique des textes, la concurrence, la décidabilité, la NP-complétude, les langages réguliers, les automates finis, les grammaires non contextuelles, les algorithmes d'apprentissage, l'algorithmique des jeux, la déduction naturelle ou encore les bases de données.

Le langage des exercices de type B était imposé : environ 50 % des exercices devaient être traités en OCaml et 50 % en C. La syntaxe SQL a, quant à elle, été évaluée via certains exercices de type A. Tous les exercices qui ont été donnés ont été entièrement ou quasiment entièrement résolus par au moins un candidat.

Les examinateurs ont pu voir d'excellentes prestations et sont satisfaits du niveau global des candidats. La majorité des candidats maîtrisent les langages OCaml et C de manière satisfaisante. Les candidats ont dans l'ensemble été très bien préparés à cette épreuve aux modalités exigeantes.

CONSEILS AUX CANDIDATS

Nous souhaitons souligner les points suivants, à l'attention des futurs candidats.

Conseils quant à la forme de l'épreuve

- Certains candidats se sont présentés devant l'examineur sans avoir leur convocation et pièce d'identité à disposition et ont ainsi perdu du temps à fouiller leur sac pour les retrouver. Nous rappelons que les formalités administratives sont incluses dans les 30 minutes de temps de passage.
- Savoir éjecter proprement une clé fait partie des compétences attendues d'un élève de MPI. Nous rappelons donc qu'il ne faut pas arracher une clé USB sans précaution, en particulier alors que des documents qui y sont présents sont encore utilisés.
- Certains candidats ont mal copié le code qu'ils avaient produit pendant la préparation sur leur clé (ou ont arraché leur clé causant des erreurs d'écriture). La responsabilité leur en incombe et ils ont dû reprendre l'exercice B depuis le début pendant le passage.
- Un sujet est composé de deux exercices et il n'est pas possible de faire de transfert de points entre les deux. Un candidat qui réussit brillamment l'exercice A et n'aborde pas l'exercice B ne pourra avoir une note supérieure à 8/20.
- Il est recommandé aux candidats de commencer par annoncer, en début d'épreuve, les questions des deux exercices qui ont été traitées en préparation. Ceci permet à l'examineur d'aider les candidats à gérer leur temps et de les interroger sur l'ensemble des questions abordées pendant la phase de préparation.
- L'examineur connaît le sujet, il n'est donc pas nécessaire pour le candidat de le réintroduire.
- Il n'est pas toujours judicieux de commencer par l'exercice A. De plus, il est possible de passer d'un exercice à l'autre pendant le passage, voire de ne pas traiter toutes les questions dans l'ordre (ou même d'en sauter quelques-unes).
- Le candidat peut aborder des questions de code (exercice de type B) non traitées en préparation pendant le passage en les programmant devant l'examineur sur la machine de la salle ou en présentant ses idées pour réaliser une telle implémentation au tableau.

- Il est nécessaire de bien lire les énoncés et en particulier celui de l'exercice B. Un nombre non négligeable de bons candidats a perdu du temps avec une lecture trop rapide et incorrecte des types imposés par un énoncé, voire en réimplémentant des fonctions qui étaient explicitement fournies dans le code compagnon, voire sans lire du tout un des énoncés.
- Il est conseillé aux candidats de bien prendre connaissance des deux exercices : certains énoncés sont difficiles à traiter à l'oral sans y avoir un peu réfléchi en amont.
- Un nombre faible mais non négligeable de candidats a été dans l'incapacité de compiler / exécuter / interpréter leur code en OCaml arguant qu'ils ne savaient utiliser aucun des outils mis à disposition. L'environnement utilisé était pourtant connu bien en amont des épreuves (voir <https://www.concours-commun-inp.fr/fr/epreuves/les-epreuves-orales.html>, rubrique MPI).
- Les examinateurs encouragent les candidats à utiliser le `Makefile` proposé pour les programmes en C, dès la phase de préparation. Lors du passage à l'oral, l'examinateur peut demander aux candidats d'utiliser ce `Makefile` à l'aide de la commande `make safe`, qui permet à l'examinateur de vérifier certains aspects de la gestion de la mémoire par le candidat. Rappelons que le contenu du `Makefile` ainsi que les consignes d'utilisation, qui sont rappelées au début de chaque exercice de type B utilisant le langage C, sont disponibles et connus en amont des épreuves.
- Le test des fonctions demandées par l'exercice B fait partie de l'évaluation et l'examinateur peut demander à voir le résultat de ces tests pendant le passage. Pour les exercices de type B en OCaml, les candidats qui interprétaient à la volée et au fur et à mesure leur code étaient avantagés sur ce point par rapport à ceux qui le compilaient, ces derniers devant effectivement se doter de fonctions d'affichage, ce qui peut leur prendre du temps (par exemple pour afficher une valeur de type `int option list`).
- Il a été constaté chez certains candidats une vitesse de frappe très lente, en particulier lors de l'utilisation de l'ordinateur pendant le passage à l'oral. Cela n'a pas été pénalisé, mais ces candidats se sont retrouvés mécaniquement désavantagés. Il est à noter que les claviers des machines sont par défaut en configuration AZERTY et que les candidats doivent y être habitués ou bien savoir en changer.

Conseils quant au fond de l'épreuve

- Les fonctionnalités des langages au programme ainsi que tout ce qui se trouve dans la rubrique "éléments de syntaxe devant être reconnus et utilisables après rappels" peuvent être librement utilisés par les candidats. Si un candidat utilise une construction des langages en dehors de ce cadre, il pourra lui être demandé d'expliquer comment répondre à la question avec uniquement les outils du programme. L'examinateur n'interdit pas, mais ne souhaite en aucun cas encourager, l'utilisation d'éléments hors programme : tous les sujets peuvent être résolus avec le fragment au programme des langages et utiliser des constructions alternatives expose le candidat à des questions qui peuvent lui faire perdre du temps sans que sa réponse ne soit davantage valorisée.
- De nombreux candidats ont été mis en difficulté par des d'erreurs et des bogues dans leurs programmes en C, davantage qu'en OCaml. S'il n'y a pas le temps dans le cadre d'une épreuve aussi courte d'utiliser des solutions de débogage complexes, il est essentiel que les candidats pensent et sachent mettre en œuvre des solutions simples pour trouver l'origine des comportements incorrects observés.
- La syntaxe SQL est généralement maîtrisée mais de nombreux candidats ont eu de grandes difficultés avec la modélisation de problèmes et le modèle entité-association. Même si aucun

formalisme n'est spécifié par le programme, des diagrammes entités-association simples font partie des attendus que les candidats doivent savoir mettre en œuvre. Il en va de même pour les schémas relationnels et leur lien avec le modèle entité-association.

- De nombreux candidats oublient de prendre en compte des cas de bases vides pour les objets construits par induction (arbres, listes, etc.).
- On observe encore de trop nombreuses confusions entre listes et tableaux en OCaml avec des tentatives non pertinentes d'accéder à un élément d'une liste via son indice.
- La structure d'arbre binaire de recherche est mal connue et parfois confondue avec celle des tas binaires. La notion d'arbres binaires de recherche équilibrés a posé de nombreux problèmes aux candidats.
- Même si elle n'est pas explicitement au programme, la fonction `Printf.printf` en OCaml a un fonctionnement très similaire à celui de la fonction `printf` en C et est particulièrement pratique pour un oral, que ce soit comme outil simple de débogage ou pour valider simplement un programme par un affichage d'exemples. Il est conseillé aux candidats de s'être familiarisés à son usage pour pouvoir l'utiliser avec aisance lorsque l'énoncé en rappelle les modalités d'usage et les codes de formatage.
- L'usage de l'opérateur OCaml `@` pour ajouter un élément en tête d'une liste est vivement déconseillé. Dans ce cas précis, elle ne permet généralement au candidat que de montrer un manque de maîtrise et de compréhension de la syntaxe OCaml.
- De nombreux candidats oublient que la pile est de taille limitée et qu'un trop grand nombre d'appels récursifs peut la faire déborder.
- Un algorithme aussi essentiel et central que le parcours de graphe n'est souvent pas maîtrisé.