

## 1/ REMARQUES GÉNÉRALES

L'épreuve d'informatique du concours CCINP proposait de découvrir des algorithmes d'intelligence artificielle appliquée à la détection d'obstacles par un sous-marin. À partir de mesures issues d'un sonar, l'algorithme doit déterminer si l'obstacle est une roche ou un objet métallique.

Le sujet permettait de balayer un large ensemble des compétences décrites dans le programme d'informatique en CPGE.

Cette session a instauré la mise en place d'un document réponse. Les candidats l'ont globalement bien utilisé. Pour les éléments de code à compléter, les correcteurs conseillent d'utiliser une encre d'une autre couleur que le noir afin de bien mettre en évidence les réponses. Pour cette deuxième session en correction numérique, les copies étaient un peu plus propres. Cependant, l'épreuve comporte un nombre de questions limité permettant au candidat de réfléchir au brouillon pour fournir une solution propre sur le document réponse. La qualité de la présentation et de la rédaction est prise en compte dans la notation.

Pour la lisibilité des programmes, il est vivement conseillé de choisir des noms de variables intelligibles, de bien soigner son indentation avec l'alignement sur les carreaux ou un trait vertical.

L'épreuve a été bien réussie par la majorité des candidats. Les correcteurs ont pu constater assez peu d'erreurs de syntaxe sur le langage Python. Attention toutefois à utiliser des variables clairement définies dans les fonctions et notamment celles passées en argument.

On retrouve cependant quelques erreurs ou maladroites : des décalages de +1 ou -1 sur les indices dans les boucles et les listes, des oublis de définition de certaines variables utilisées dans les fonctions, une représentation des nombres très mal maîtrisée. Des confusions entre les listes et les tableaux numpy restent persistantes avec des mélanges de méthodes entre les deux types. La confusion entre l'indice et l'élément dans une boucle : « for i in temps » puis un accès à « temps[i] ».

## 2/ REMARQUES SPÉCIFIQUES

**Q1** : cette question qui consistait à créer un tableau des différents instants de la simulation, a été plutôt mal traitée : confusion entre arange et linspace (pas versus nombre de valeurs, dernière valeur exclue versus incluse par défaut) ; même la solution en remplissant une liste est souvent fautive à cause de problèmes d'arguments dans la boucle for.

**Q2** : beaucoup de candidats ne prennent pas en compte la définition de la fonction sur deux intervalles faisant intervenir un test sur le temps. Il y a encore beaucoup d'erreurs d'initialisation de la valeur à retourner : des array([]) sur lesquels le candidat fait des append ; des empty(N) pour lesquels le candidat ne remplit pas toutes les cases. Les correcteurs préfèrent l'utilisation de la fonction zeros() qui initialise la variable avec des zéros partout.

**Q3** : pratiquement aucun candidat ne met les bons arguments dans la fonction et notamment le premier argument. La plupart des candidats appliquent la transformée sur le temps, d'autres recopient simplement les arguments de la documentation...

**Q4** : cette question d'analyse a été moyennement bien traitée. Peu de candidats arrivent à conclure que la méthode permet de s'affranchir des bruits de mesure et pas que le spectrogramme effectue le traitement du bruit.

**Q5** : question bien traitée par les candidats. Pour la lisibilité du test, il est intéressant de définir des variables intermédiaires.

**Q6** : cette question est plutôt bien traitée. Certains candidats oublient toutefois des morceaux de la formule, dt et df notamment. On peut noter que faire le produit dt.df après la boucle permet de gagner en efficacité.

**Q7** : cette question est assez mal traitée. Peu de candidats trouvent la fonction affine à appliquer au tableau P. Pour ceux qui y arrivent, souvent le calcul du min et du max est laissé dans la boucle augmentant fortement la complexité de l'algorithme. Certains candidats ne renvoient pas le résultat dans le type demandé.

**Q8** : les candidats répondant à cette question l'ont plutôt réussie.

**Q9** : une grande majorité des candidats ne connaissent pas l'espace nécessaire pour stocker un réel en double précision. On retrouve des confusions entre les bits et les octets. Il y aura toujours des questions de ce genre dans les épreuves, les correcteurs s'attendent à un minimum de connaissances sur la représentation des informations en mémoire.

**Q10** : question bien traitée par les candidats.

**Q11** : question généralement bien traitée par les candidats quand il n'y a pas de confusion sur les indices des colonnes.

**Q12** : question où la principale difficulté consistait à vérifier que la nouvelle valeur aléatoire n'était pas déjà présente dans la liste construite. Cette vérification est souvent très mal réalisée. L'utilisation de « a in Liste » (ou « a not in Liste ») était plus appropriée.

**Q13** : question plutôt bien traitée. La principale erreur a été de mettre la valeur de la donnée (donnees[i][ind]) dans les listes droite ou gauche plutôt que la donnée complète (donnee[i]).

**Q14** : question assez bien traitée. Des confusions entre les variables taille et n\_donnees.

**Q15** : question assez bien traitée.

**Q16** : question plutôt mal traitée. La difficulté était de bien comprendre ce que faisait la fonction et comment y parvenir. Il fallait pour cela prendre bien le temps nécessaire.

**Q17** : question d'analyse bien traitée.

**Q18** : question rarement traitée par les candidats. Ceux ayant compris ce qu'il fallait faire oublient souvent les return des appels récursifs.

**Q19** : question très rarement traitée. Les candidats ayant répondu, l'ont généralement assez bien fait.

**Q20** : question d'analyse bien traitée.