

1/ COMMENTAIRE GÉNÉRAL SUR L'ÉPREUVE

L'épreuve d'informatique du concours CCINP proposait de travailler sur la reconnaissance de caractères. La première partie traitait de l'acquisition du document en format numérique. La deuxième partie traitait de la rotation automatique du document, détection automatique des lignes, puis restauration des caractères bruités avant leur détection. La troisième partie traitait de la détermination des caractères par une méthodes d'intelligence artificielle.

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

Cette session, qui est la première pour l'épreuve d'informatique de tronc commun en filière PC, a été commune avec celle de la filière PSI. Comme pour cette dernière, la composition s'est faite à l'aide d'un document réponse que les candidats ont globalement bien utilisé. Il est rappelé qu'il est utile de réfléchir sur un brouillon afin de porter le plus proprement possible la solution sur le document.

La propreté des copies était bonne. 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'utilisation des commentaires, bien qu'appréciée, ne doit pas surcharger la copie et faire perdre trop de temps aux candidats. Certains bons candidats ont énormément commenté leur code mais cela ne leur a pas permis d'aller très loin dans le sujet.

La gestion des parenthèses est souvent mal maîtrisée : oubli pour les fonctions (par exemple `range n :`) ; oubli dans les expressions mathématiques ($d^{**1/2}$ est différent de $d^{**(1/2)}$).

On trouve aussi des syntaxes `L.append(a,b)` pour ajouter deux éléments à une liste.

Même si le jury est parfois « souple » sur l'indentation, certaines erreurs d'indentation conduisent à des algorithmes faux.

2/ ANALYSE DÉTAILLÉE DES QUESTIONS

Q1 : Cette question a été assez mal traitée. Le calcul demandé pouvait être fait en ordre de grandeur, mais beaucoup d'erreurs ont été constatées.

Q2 : Des erreurs de typographie sont malheureusement restées dans le programme donné en exemple. Il fallait remplacer `p` par `n0` et `q` par `n1`... D'autre part, la syntaxe `img[i,j]` est à remplacer par `img[i][j]`. Le jury a évidemment été souple sur les réponses faites par les candidats en acceptant les différentes interprétations possibles.

Cependant, beaucoup de candidats se contentent de dire "double boucle" donc $O(n^2)$! Mais qui est n ? Une phrase pour dire que les instructions dans la deuxième boucle était en temps constant était également bienvenue.

Q3 : Beaucoup de copies ont modifié l'image en entrée plutôt que d'en créer une nouvelle, l'énoncé n'étant pas explicite à ce sujet cela n'a pas été pénalisé, de même que les copies faisant un mélange entre (p, q) et (n_0, n_1) suite à l'erreur d'énoncé de la question précédente. En revanche, les copies qui effectuaient une conversion en niveaux de gris, que ce soit manuellement en extrayant trois valeurs par pixel de l'image d'entrée ou via `conversion_gris()` l'ont été. Dans cette question ainsi que dans plusieurs autres, certaines copies font le choix de ne pas faire appel à `dimension()` pour obtenir celles de l'image, ce n'était naturellement pas pénalisé tant que le code fonctionnait mais si l'énoncé offre des facilités, autant les utiliser.

Q4 : Cette question est bien traitée.

Q5 : Cette question a été moyennement bien traitée. Les candidats ont eu un peu de mal à exploiter la fonction partiellement donnée.

Q6 : Cette question a été assez mal traitée. Il était attendu que se restreindre à 8 bits non signés permettait d'avoir un gain en terme de stockage par rapport à une représentation en nombre flottant par exemple. Et surtout, la deuxième partie de la question, un nombre négatif ne peut pas être représenté avec un nombre non signé. Le calcul donnera une représentation associée à un nombre positif (pas la valeur absolue contrairement à ce que beaucoup pensent) rendant le niveau de gris associé incorrect (figure 5 du sujet).

Q7 : Cette question a été globalement bien traitée. Les erreurs d'indentations et la position de la mise à zéro du compteur étaient ici particulièrement critiquées.

Q8 : Cette question a été assez mal traitée. Il est vrai que la gestion du drapeau (`deb = -1`) pour déterminer si on est ou non à l'intérieur d'un bloc correspond à une manière de penser à laquelle certains candidats n'ont pas forcément été réceptifs, mais chercher à comprendre et à s'adapter à des codes écrits par d'autres fait aussi partie des objectifs de la formation.

Q9 : Cette question a été souvent traitée. Si le nom de la dichotomie a été assez largement donné, la justification de l'arrêt de l'algorithme a parfois été farfelue : "l'algorithme s'arrête car il y a un while avec un epsilon" n'est pas une réponse ! De trop nombreuses copies assurent la terminaison de l'algorithme par la décroissance de la taille de l'intervalle de recherche ou par la taille finie de celui-ci sans plus de précisions (mais aussi souvent pour des raisons pires, comme l'existence d'un unique maximum dans l'intervalle — le bon conditionnement mathématique du problème n'implique pas la terminaison d'un algorithme visant à le résoudre). On rappelle qu'une suite décroissante (même strictement, ce qui était rarement indiqué de toutes façons) et positive ne tend pas forcément vers 0 : il ne faut pas confondre ce cas avec celui d'un compteur entier qu'on décrémenterait par exemple. Le nombre d'itérations quand il a été calculé a souvent été correct.

Q10 : Cette question a été assez bien traitée. Toutefois, le calcul du milieu des segments est assez souvent faux : $(b-a)/2$, $a*b$, ... alors même que la formule était donnée pour le point c juste au-dessus.

Q11 : Certains candidats font vraiment preuve de beaucoup d'imagination pour rendre le graphe illisible. Ne pas hésiter à utiliser un brouillon !

Q12 : Cette question est assez bien traitée par le peu de candidats ayant abordé la partie sur les graphes.

Q13 : Question plutôt mal traitée. Cette question nécessitait de bien comprendre l'algorithme expliqué. Une ambiguïté dans le choix du chemin était restée dans le sujet. Il fallait comprendre qu'on choisit le plus court chemin en nombre d'arêtes. En cas d'égalité en nombre d'arêtes, on choisit celui dont la somme des étiquettes est la plus faible. Toutefois, l'algorithme fonctionnait quand même en comprenant le plus court en somme d'étiquette, mais nécessitait plus d'étapes. Le jury a noté juste toute solution qui suivait la logique comprise par le candidat...

Q14 : Il fallait calculer la coupe sur le résultat de la Q13.

Q15 : Il fallait clairement identifier les ensembles A et B aux pixels noirs et blancs de l'image finale (une inversion entre les deux couleurs n'était pas pénalisée). De nombreuses copies ont essayé de faire le lien

avec les pixels « gris clairs » ou « gris foncés » alors même que le sens de cet algorithme était de ne plus faire une partition en fonction d'un seuil, ce qui a semble-t-il été mal compris par beaucoup.

Q16/Q17/Q18 : Les questions de SQL ont donné un second souffle à beaucoup de candidats. Les questions sont assez bien traitées. On notera toutefois les erreurs suivantes :

- attribut.TABLE au lieu de TABLE.attribut ;
- BETWEEN mal maîtrisé auquel on pourra préférer une double comparaison ;
- dans les jointures, bien lier les bons identifiants ;
- beaucoup de WHERE après le GROUP BY.

Q19 : Cette question n'a pas été aussi bien réussie que ce qu'elle aurait dû être. Il convient d'être précis quant au type des valeurs renvoyées. Cela est certainement dû à de la précipitation, la fin de l'épreuve étant sans doute proche !

Q20 : On notera une maîtrise assez approximative des dictionnaires. Très très peu de candidats pensent à initialiser le dictionnaire si la clé n'est pas présente et font directement un append sur un objet inexistant.

Q21 : Question souvent bien traitée. Beaucoup de copies ont oublié l'existence de la fonction auxiliaire `dimension` à ce stade, mais ce n'était pas forcément dérangeant. En revanche, les copies calculant la racine carrée à chaque itération à l'intérieur des boucles (parfois sur la somme partielle, parfois directement sur l'élément qui venait d'être élevé au carré...) n'ont pas été valorisées. Concernant le calcul de la racine, l'utilisation de `sqrt` était autorisée même si aucune attention n'était prêtée à sa disponibilité.

Q22 : Cette question a été assez peu traitée. La structure passée en entrée a été mal comprise, beaucoup de copies ne considérant qu'une seule image par symbole. Il fallait à nouveau faire attention à savoir si la clé était déjà présente dans le dictionnaire ou non, une initialisation systématique à chaque nouvelle clé rencontrée puis une itération des images correspondantes fonctionnait bien.

Q23 : Aucune connaissance spécifique d'un algorithme de tri n'étant exigible au programme, nommer un algorithme et la complexité correspondante (nécessaire) suffisait pour obtenir des points. Les réponses indiquant — correctement — un tri de complexité meilleure que quadratique au moins dans le cas moyen (tri fusion, tri rapide...) étaient valorisées en sus.

Q24 : Question très mal traitée. Le sujet aurait certainement dû être plus clair sur la méthode en expliquant le tri par insertion.

Q25 : Question peu traitée et dépendante de la question précédente.

Q26 : Très peu de candidats se sont lancés dans cette question, mais certains l'ont fait correctement et avec élégance.

Q27 : Une très grande partie des candidats a analysé les résultats obtenus et fourni une conclusion adéquate.