

Rapport de Correction - Examen ASD S1 2025-2026

Fichier Source : doc20260115223319.pdf Nombre de Copies : 7

COPY NUMBER: 1

Analyse :

Algorithmique :

- Boucle while (`N > 44`). Condition absurde et "magique".
- `n` est utilisé pour la lecture de `N` ?? Non `scanf("%d" , &N)`. `n` lu dans la boucle mais écrasé par `n = 0`.
- Confus, utilisation de constantes magiques.
- Termes incorrects (abzent).

NOTE FINALE : 03 / 20

Feedback :

- Appréciation globale : Très Insuffisant.
-

COPY NUMBER: 2

Analyse :

Algorithmique :

- Lecture OK.
- Boucle `for`.
- Calculs finaux dans la boucle : `sP = N - x ??` `x` est le nombre de séances de l'étudiant, pas le nombre de présents ! Grosse confusion sur la signification de `x`.
- Condition d'arrêt `if (x == N || x == S)`. Encore confusion entre `x` (séances d'un étudiant) et les compteurs globaux.
- Ne compte rien (variables écrasées à chaque tour).

NOTE FINALE : 06 / 20

Feedback :

- **Appréciation globale : Insuffisant.** Confusion totale sur les variables.
-

COPY NUMBER: 3

Analyse :

Algorithmique :

- Lecture OK.
- Boucle `for`. Logique interne OK.
- Compteurs Absence, Presence.
- Arrêt : `if (i == N || Absence == S)`. Utilise `break` (interdit) ou `return`.
- Affiche status final DANS la boucle (à chaque itération si condition remplie).

NOTE FINALE : 14 / 20

Feedback :

- **Appréciation globale : Moyen / Bon.** Pénalité pour `break` et structure "tout dans la boucle".
-

COPY NUMBER: 4

Analyse :

Algorithmique :

- Pas de lecture initiale de N ! (Variable non init).
- Boucle `for (i=1; i<=N)`. N aléatoire.
- Lecture de N **DANS** la boucle !?
- Modification de i (`i = i + 1`) dans le `for` (saut des étapes).
- `absent = N - present`. present non init.

NOTE FINALE : 04 / 20

Feedback :

- **Appréciation globale : Très Insuffisant.**
-

COPY NUMBER: 5

Analyse :

Algorithmique :

- Lectures OK.
- Boucle `for`.
- Logique interne OK.
- Arrêt : `if (absent >= S) i = N;`. (Hack de sortie, ok).
- Condition finale inversée : `if (absent >= S) -> Valid ?? Non, cancelled !` (Logique valid si `absent >= S` est faux).

NOTE FINALE : 14 / 20

Feedback :

- **Appréciation globale : Moyen.** Attention à la condition finale (inversée).
-

COPY NUMBER: 6

Analyse :

Algorithmique :

- Boucle while ($(L) < S \ \&\& \ i < N$). Condition composée parfaite. $L = absents$.
- Logique interne simple et correcte.
- Affichage final correct.

NOTE FINALE : 18 / 20

Feedback :

- Appréciation globale : Très Bon.
-

COPY NUMBER: 7

Analyse :

Algorithmique :

- Boucle `for. i <= n` (utilise `n` lu pour nombre étudiants).
- Condition arrêt `if (S1 <= S)` (Absents <= Seuil).
- Sinon `else { n = i; }`. Hack de sortie (met fin à la boucle for car `i` va dépasser `n` ? Non, `n=i` rend `i <= n` vrai tant que `i` n'augmente pas. `i++` fera `i > n` au tour suivant. Astucieux mais `n` est écrasé, donc on perd le nombre total.
- Pas critique car `i` contient le nombre traité.

NOTE FINALE : 15 / 20

Feedback :

- **Appréciation globale : Bon.**