

Rapport de Correction - Examen ASD S1 2025-2026

Fichier Source : doc20260115220910.pdf **Nombre de Copies :** 18

COPY NUMBER: 1

Analyse :

Algorithmique :

- Lecture correcte de N, A, S.
- Lecture de x **AVANT** la boucle while. La variable x ne change jamais dans la boucle ! Boucle infinie si $x \geq A$ ou sautée si $x < A$.
- Corps de boucle : $a = N - p$. Logique confuse.
- Condition d'arrêt while ($x \geq A$) n'a aucun sens par rapport à l'énoncé (fin si tous traités ou seuil absents).

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Initialisation	3 / 3	OK.
Condition boucle	0 / 4	Incorrecte et boucle potentiellement infinie.
Logique prés./abs.	2 / 4	Test correct, mais mal placé (hors itération).
Compteurs	1 / 3	Incrémentation présente mais logique fausse.
Affichages inter.	1 / 2	Présents mais données fausses.
Affichage final	1 / 1	OK.

NOTE FINALE : 11 / 20

Feedback :

- **Appréciation globale : Moyen -**. Erreur classique de débutant : lecture de la variable de contrôle hors de la boucle.

COPY NUMBER: 2

Analyse :

Algorithmique :

- Lecture X dans la condition du while ? `if (scanf(...)) <= A)`. C'est astucieux mais techniquement `scanf` renvoie le nombre d'éléments lus, pas la valeur ! Donc `scanf` renvoie 1, qui est comparé à A. Si A > 1, résultat toujours vrai.
- Boucle `while (i <= N || j <= S)` : OU logique au lieu de ET. Continue tant qu'une limite n'est pas atteinte (risque dépassement).
- Calculs finaux `b = N - i` corrects sur le principe.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	0 / 3	Absente (un <code>printf</code> demande mais pas de <code>scanf</code> pour N, A, S !!). Variables init à 0 dans déclarations pseudo <code>printf(N, A, S)</code> .
Initialisation	2 / 3	Variables init.
Condition boucle	2 / 4	Erreur logique .
Logique prés./abs.	1 / 4	Erreur majeure sur utilisation de <code>scanf</code> en expression.
Compteurs	2 / 3	OK.
Affichages inter.	2 / 2	OK.
Affichage final	1 / 1	OK.

NOTE FINALE : 08 / 20

Feedback :

- **Appréciation globale : Insuffisant.** Pas de lecture des paramètres initiaux. Confusion sur le retour de `scanf`.

COPY NUMBER: 3

Analyse :

Algorithmique :

- Boucle `while (i <= N, i < S, i++)` : Syntaxe `for` utilisée dans un `while`? Opérateur virgule en C évalue tout et renvoie le dernier. Condition `i++` est toujours vraie (sauf overflow). Boucle infinie probable ou syntax error.
- Variables non initialisées `N, A, S`. Pas de lecture.
- Logique interne correcte (`if (x < A)`).

Notation :

Critère	Points	Commentaire
Lecture <code>N, A, S</code>	0 / 3	Absente.
Initialisation	0 / 3	Variables non init.
Condition boucle	0 / 4	Syntaxe invalide.
Logique prés./abs.	3 / 4	Correcte.
Compteurs	0 / 3	Calculs mathématiques au lieu d'incrémantion.
Affichages inter.	1 / 2	OK.
Affichage final	1 / 1	OK.

NOTE FINALE : 05 / 20

Feedback :

- **Appréciation globale : Très Insuffisant.** Code non fonctionnel.

COPY NUMBER: 4

Analyse :

Algorithmique :

- Lectures correctes.
- Boucle `for` sur N. Manque condition d'arrêt sur le seuil S.
- Logique interne correcte.
- Mais **aucun compteur** n'est incrémenté ! `student_absent` utilisé à la fin n'est jamais calculé.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Initialisation	1 / 3	Manque compteurs.
Condition boucle	2 / 4	Partielle (S ignoré).
Logique prés./abs.	3 / 4	Bon test.
Compteurs	0 / 3	Absents.
Affichages inter.	1 / 2	OK.
Affichage final	0 / 1	Variable inexistante.

NOTE FINALE : 10 / 20

Feedback :

- **Appréciation globale : Moyen -**. Structure ok mais oubli fonctionnel majeur (les compteurs).

COPY NUMBER: 5

Analyse :

Algorithmique :

- Lectures correctes.
- Boucle `for` sur N.
- Logique correcte.
- Compteurs incrémentés mais **non initialisés** (Absent, Present).
- Condition d'arrêt sur S manquante dans la boucle (testée seulement à la fin).

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Initialisation	0 / 3	Non initialisées.
Condition boucle	2 / 4	Partielle.
Logique prés./abs.	4 / 4	OK.
Compteurs	3 / 3	OK (sauf init).
Affichages inter.	1 / 2	OK.
Affichage final	1 / 1	OK.

NOTE FINALE : 14 / 20

Feedback :

- **Appréciation globale : Moyen.** Attention à l'initialisation des variables locales en C (valeur indéterminée).

COPY NUMBER: 6

Analyse :

Algorithmique :

- Lectures correctes.
- Boucle for : return 0 à la fin de la première itération. Le programme s'arrête tout de suite.
- Utilise absent_count non déclaré.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Condition boucle	1 / 4	Interrompue.
Logique prés./abs.	2 / 4	OK.
Compteurs	0 / 3	Nom variable incorrect.
Affichage final	0 / 1	-

NOTE FINALE : 06 / 20

Feedback :

- Appréciation globale : Fragile.
-

COPY NUMBER: 7

Analyse :

Algorithmique :

- Boucle while (*i* <= *N*) mais *i* n'est **jamais initialisé** (valeur poubelle) ni incrémenté. Boucle possiblement infinie ou jamais exécutée.
- Lectures dans la boucle ? `scanf("%d", &absents)` -> l'utilisateur doit saisir le nombre d'absents ? Contresens de l'énoncé (on doit le calculer).

NOTE FINALE : 04 / 20

Feedback :

- **Appréciation globale : Très Insuffisant.** Incompréhension de l'énoncé.
-

COPY NUMBER: 8

Analyse :

Algorithmique :

- `break` utilisé (Interdit par l'énoncé mais fonctionnel).
- Logique correcte.
- Affichages complets.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Initialisation	3 / 3	OK.
Condition boucle	3 / 4	Utilise <code>break</code> (interdit) au lieu condition composée.
Logique prés./abs.	4 / 4	OK.
Compteurs	3 / 3	OK.
Affichages inter.	2 / 2	OK.
Affichage final	1 / 1	OK.

NOTE FINALE : 19 / 20

Feedback :

- **Appréciation globale : Très Bon.** Code robuste. Pénalité minime pour le `break`.

COPY NUMBER: 9

Analyse :

Algorithmique :

- Initialisation `na=0` mais `np` non initialisé.
- `break` utilisé.
- Logic `if (i == N || na == S)`.
- Globalement ok sauf init `np`.

NOTE FINALE : 16 / 20

Feedback :

- Appréciation globale : Bon.
-

COPY NUMBER: 10

Analyse :

Algorithmique :

- Logique correcte.
- Utilise compteurs B, P.
- Condition `if (X < S)` pour annulation ? Devrait être B (compteur absents) < S. Confusion variable.

NOTE FINALE : 13 / 20

Feedback :

- **Appréciation globale : Moyen.** Confusion sur la condition finale.
-

Analyse :

Algorithmique :

- Boucle while ($i \leq N$) avec i non initialisé.
- Pas de lecture de x (séances).
- Comparaisons incohérentes.

NOTE FINALE : 03 / 20

Feedback :

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

COPY NUMBER: 12

Analyse :

Algorithmique :

- Condition while utilise count == 5. Pourquoi 5 ? Constante magique au lieu de s.
- Initialisation student=1 OK.
- Logique correcte.

NOTE FINALE : 14 / 20

Feedback :

- Appréciation globale : Moyen / Bon.
-

COPY NUMBER: 13

Analyse :

Algorithmique :

- Boucle `for` sur N. Pas d'arrêt sur S.
- Logique correcte.
- Compteurs ok.

NOTE FINALE : 12 / 20

Feedback :

- Appréciation globale : Moyen.
-

COPY NUMBER: 14

Analyse :

Algorithmique :

- Lectures correctes.
- Saisie S après la boucle !? (Impossible d'arrêter avant).
- Logique $m < S$ finale correcte.

NOTE FINALE : 12 / 20

Feedback :

- **Appréciation globale : Moyen.** Erreur ordre instructions.
-

COPY NUMBER: 15

Analyse :

Algorithmique :

- Condition while (*i* <= *N* || *Sum1* == *S*). || au lieu de && ? (Tant que pas fini OU seuil atteint... non ça devrait être && non atteint). Avec ||, si *S* atteint mais *i* < *N*, ça continue.
- Mais correct sur le reste.

NOTE FINALE : 15 / 20

Feedback :

- Appréciation globale : Bon.
-

COPY NUMBER: 16

Analyse :

Algorithmique :

- Déclaration `for (A=1; A<=N; A++)` : Utilise A (seuil min) comme compteur de boucle ! Écrase la valeur de A. Catastrophe.
- `count` non déclaré.

NOTE FINALE : 05 / 20

Feedback :

- **Appréciation globale : Très Insuffisant.** Erreur grave de réutilisation de variable.
-

COPY NUMBER: 17

Analyse :

Algorithmique :

- Condition `if (i <= N && OS < S)` à l'intérieur de la boucle `for` ? Redondant mais ok.
- Calcul `n = n + i` (somme des indices ?). Inutile.
- Variables OS, PS ok.

NOTE FINALE : 14 / 20

Feedback :

- Appréciation globale : Moyen / Bon.
-

COPY NUMBER: 18

Analyse :

Algorithmique :

- Variable `n` déclarée dans la boucle (interdit en C89 strict mais ok C99).
- Logique correcte.
- Condition arrêt `if (N student processed || ...)` pseudo-code.
- Condition finale `if (OS = s)` (affectation au lieu de comparaison `==`).

NOTE FINALE : 10 / 20

Feedback :

- **Appréciation globale : Moyen -**. Erreurs de syntaxe C (`=`).