

# Rapport de Correction - Examen ASD S1 2025-2026

Fichier Source : doc20260115221134.pdf Nombre de Copies : 22

---

Analyse :

Algorithmique :

- Lecture N, A, S correcte.  
Boucle `for (int i = 0; i <= N || i == S; i++) :`
  - Condition d'arrêt incorrecte (`i == S` au lieu de tester le nombre d'absents).
  - Condition `i <= N` correcte mais indexation à 0 finit à N (soit N+1 itérations).
- Déclaration `if (int x < A)` à l'intérieur du `if` : Interdit ! `int x` déclare une nouvelle variable locale non initialisée qui masque tout. De plus, `x` n'a pas été lu (`scanf` manquant dans la boucle).
- `i = S` dans le `if` : Tentative de forcer la sortie de boucle ?
- `con` non initialisé. `con` utilisé mais non déclaré.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Initialisation	0 / 3	Compteurs non initialisés/déclarés incorrectement.
Condition boucle	1 / 4	Condition d'arrêt fausse.
Logique prés./abs.	0 / 4	Erreur grave : redéclaration <code>int x</code> dans le <code>if</code> sans saisie.
Compteurs	0 / 3	Noms variables incohérents ( <code>con</code> vs <code>con</code> ).
Affichages inter.	1 / 2	Texte sans valeurs.
Affichage final	0 / 1	-

NOTE FINALE : 05 / 20

Feedback :

- Appréciation globale : Très Insuffisant. Problèmes de portée de variables et de logique de base.

**Analyse :****Algorithmique :**

- Syntaxe `printf(N, A, S)` incorrecte (pas de format). `scanf(N, A, S)` incorrect (pas de format ni &).
- Initialisation `int N=0` effacée par le `scanf` (si ça marchait).
- Boucle `for (i=1; i<=N)`.
- Lecture `x` deux fois ?? `scanf("%x")` (hexadécimal ??) puis `scanf("%d", &x)`.
- Condition d'arrêt `if (absent == S) i = N;` (Astuce pour sortir, fonctionnelle).

**Notation :**

Critère	Points	Commentaire
Lecture N, A, S	0 / 3	Syntaxe <code>scanf</code> totalement fausse (pas de %d).
Initialisation	3 / 3	OK.
Condition boucle	2 / 4	Arrêt simulé via <code>i=N</code> . Ok.
Logique prés./abs.	2 / 4	Double saisie confuse. <code>scanf("%x")</code> .
Compteurs	3 / 3	OK.
Affichages inter.	2 / 2	OK.
Affichage final	1 / 1	OK.

**NOTE FINALE : 13 / 20****Feedback :**

- **Appréciation globale : Moyen.** La logique est là, mais la syntaxe des E/S est à revoir d'urgence (`scanf`).
-

Analyse :

Algorithmique :

- Lectures correctes.
- Boucle `for` : Condition `B == S` étrange pour *continuer* ? Devrait être `B < S`.
- Compteur `count` utilisé mais non déclaré. `P` et `B` déclarés mais non initialisés (et utilisés ensuite).
- Confusion entre `count`, `P`, `B`.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Initialisation	0 / 3	Variables non initialisées.
Condition boucle	1 / 4	Incorrecte.
Logique prés./abs.	2 / 4	Structure if ok, mais variables inconnues.
Compteurs	0 / 3	<code>count</code> non déclaré.
Affichage final	0 / 1	-

NOTE FINALE : 06 / 20

Feedback :

- **Appréciation globale : Fragile.** Code non compilable.
-

**Analyse :**

**Algorithmique :**

- Lectures correctes.
- Lecture de  $x$  **avant** la boucle (une seule fois pour tous).
- Si  $x \geq A$ , boucle `for` imbriquée qui affiche "present" N fois de suite avec la même valeur ? Logique totalement fausse.
- Pas de traitement par étudiant.

**Notation :**

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Condition boucle	0 / 4	Logique aberrante (boucle dans le if).
Logique prés./abs.	1 / 4	Test sur valeur unique.
Compteurs	0 / 3	-
Affichages inter.	0 / 2	-
Affichage final	0 / 1	-

**NOTE FINALE : 04 / 20**

**Feedback :**

- **Appréciation globale : Très Insuffisant.** Algorithme non compris.
-

**Analyse :**

**Algorithmique :**

- Lectures correctes.
- Lecture de  $x$  **avant** la boucle.
- Boucle `for (i=0; i<X; i++)` : Itère  $x$  fois (nombre de séances) au lieu de  $N$  fois (nombre d'étudiants). Contresens complet.
- La boucle répète juste le statut de l'étudiant unique  $x$  fois.

**NOTE FINALE : 05 / 20**

**Feedback :**

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

Analyse :

- Algorithmique :**
- Boucle `do { for ... } while` complexe et inutile.
  - Boucle `for` sur `N` avec lecture interne correcte.
  - Le `do-while` extérieur dépend de `H < S` (seuil). C'est une façon originale de gérer l'arrêt, mais le `for` interne va quand même traiter tout le monde avant de vérifier le `while`. Donc arrêt tardif.
  - Logique interne correcte.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Initialisation	3 / 3	OK.
Condition boucle	2 / 4	Structure boucle imbriquée maladroite ne permettant pas l'arrêt immédiat au seuil.
Logique prés./abs.	4 / 4	OK.
Compteurs	3 / 3	OK.
Affichages inter.	2 / 2	OK.
Affichage final	1 / 1	OK.

NOTE FINALE : 18 / 20

Feedback :

- **Appréciation globale : Bon.** Structure un peu lourde mais fonctionne.
-

Analyse :

Algorithmique :

- Variable `n` déclarée mais utilise `n` non init dans `for`. Ah, `n` est le paramètre lu ? Non, `scanf("%d", &N)`. Utilise `N` et `n` de façon interchangeable ? Code: `i <= n`. `n` n'est pas init.
- Utilise `x` (minuscule) dans `scanf`, `x` non déclaré ? Ou l'inverse.
- Utilise `k` et `j` pour les compteurs.
- Logic `if (x < A)` correcte.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	2 / 3	Confusion de noms de variables ( <code>n</code> vs <code>N</code> ).
Initialisation	1 / 3	<code>n</code> utilisé dans boucle non initialisé.
Condition boucle	1 / 4	Boucle risque de ne pas s'exécuter ( <code>n</code> ?). Pas d'arrêt sur seuil.
Logique prés./abs.	3 / 4	OK.
Compteurs	3 / 3	OK.
Affichage final	1 / 1	OK.

NOTE FINALE : 11 / 20

Feedback :

- **Appréciation globale : Moyen -.** Attention à la casse et aux noms de variables.



**Analyse :****Algorithmique :**

- Lecture OK.
- Boucle `while` bien construite (`i <= N && Z != S`).
- Lecture `x` dans la boucle.
- Logique `if (A > X)` (Absent) OK.
- Problème : `i++` est dans le `else` (présent) uniquement ! Si l'étudiant est absent, on n'incrémente pas `i`. Boucle potentiellement infinie si que des absents (mais arrêt sur `Z != S`). Par contre, si l'étudiant est absent, on reste sur le même `i` pour le suivant ? Logique étrange. On devrait incrémenter `i` tout le temps.

**Notation :**

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Condition boucle	4 / 4	Condition composée correcte.
Logique prés./abs.	2 / 4	Erreur : l'incréméntation de <code>i</code> est conditionnelle (seulement si présent).
Compteurs	3 / 3	OK.
Affichage final	1 / 1	OK.

**NOTE FINALE : 13 / 20****Feedback :**

- **Appréciation globale : Moyen.** L'étudiant ne passe au suivant que s'il est présent ?
-

**Analyse :**

**Algorithmique :**

- Condition `if (X < A)` placée **avant** la boucle et avant lecture de `x`.
- Boucle `for` sur `N`.
- Corps de boucle : `if (N == A || N++)`. N'a aucun sens. Ne lit rien.

**NOTE FINALE : 02 / 20**

**Feedback :**

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

Analyse :

- Algorithmique :**
- Lecture OK.
  - Boucle `do ... while`. Condition `NS < S`. Correct.
  - Corps : lecture et tests OK.
  - Affichages : OK.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Initialisation	3 / 3	OK.
Condition boucle	4 / 4	<code>do while</code> bien utilisé.
Logique prés./abs.	4 / 4	OK.
Compteurs	3 / 3	OK.
Affichages inter.	2 / 2	OK.
Affichage final	1 / 1	OK.

NOTE FINALE : 20 / 20

Feedback :

- **Appréciation globale : Très Bon.**
-

**Analyse :**

**Algorithmique :**

- Condition boucle `for (i = 1; i <= N || S > N; i++)`. Le `|| S > N` rend la condition toujours vraie si `S > N` (ce qui est possible). Boucle infinie.
- `Sum1 = Sum1 + Suma`. `Suma` n'est pas déclaré. Voulait dire `1` ? ou `Sum1` ?
- Logique de comptage fausse.

**NOTE FINALE : 07 / 20**

**Feedback :**

- **Appréciation globale : Insuffisant.** Code non fonctionnel (variables inconnues).
-

**Analyse :**

**Algorithmique :**

- Utilisation de `#define` pour N, A, S. L'énoncé disait "Lire N, A, S". (-3 pts lecture).
- Boucle `while` correcte.
- Logique correcte.

**Notation :**

Critère	Points	Commentaire
Lecture N, A, S	0 / 3	Utilisation de constantes au lieu de <code>scanf</code> . Hors sujet.
Initialisation	3 / 3	OK.
Condition boucle	4 / 4	OK.
Logique prés./abs.	4 / 4	OK.
Compteurs	3 / 3	OK.
Affichages inter.	2 / 2	OK.
Affichage final	1 / 1	OK.

**NOTE FINALE : 17 / 20**

**Feedback :**

- **Appréciation globale :** Très Bon code, mais attention à bien lire l'énoncé (entrées dynamiques demandées).
-

**Analyse :**

**Algorithmique :**

- Lecture OK.
- Boucle `while (n > S || A == 5)`. Condition totalement arbitraire et fausse.
- Condition ne change pas dans la boucle (si ce n'est `n` qui n'est pas décrémenté, ah si `total sum` ?).
- Boucle infinie.

**NOTE FINALE : 05 / 20**

**Feedback :**

- **Appréciation globale : Insuffisant.**
-

**Analyse :**

**Algorithmique :**

- Lecture de x avant la boucle.
- Pas de boucle !! Code séquentiel.
- `absent = n + 1`. Formule magique ?

**NOTE FINALE : 03 / 20**

**Feedback :**

- **Appréciation globale : Très Insuffisant.** Pas d'algorithme.
-

Analyse :

- Algorithmique :**
- Boucle `for` sur `N`.
  - A l'intérieur, `if (absent < S)`. C'est une façon valide de gérer l'arrêt (ne plus rien faire si seuil atteint).
  - Logique correcte.
  - Affichage `Student number: %02d`. Soigné.

Notation :

Critère	Points	Commentaire
Lecture <code>N</code> , <code>A</code> , <code>S</code>	3 / 3	OK.
Condition boucle	3 / 4	Arrêt géré par un <code>if</code> englobant. Fonctionnel.
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.**
-



**Analyse :**

**Algorithmique :**

- Boucle `while (i <= N && absent < S)`. Correcte.
- Logique parfaite.
- Affichage final correct.

**NOTE FINALE : 20 / 20**

**Feedback :**

- **Appréciation globale : Excellent.**
-

Analyse :

- Algorithmique :
- Code minimaliste (commentaires `// read A` au lieu du code).
  - Logique `while` correcte.
  - Incomplet sur la lecture.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	0 / 3	Commentaires au lieu de code.
Initialisation	3 / 3	OK.
Condition boucle	4 / 4	OK.
Logique prés./abs.	3 / 4	⌘ non lu explicitement (implicite).
Compteurs	3 / 3	OK.
Affichage final	1 / 1	OK.

NOTE FINALE : 14 / 20

Feedback :

- **Appréciation globale : Moyen.** Code fonctionnel mais les E/S sont simulées par des commentaires.

**Analyse :**

**Algorithmique :**

- Boucle `for`.
- Logique interne : `if (x < A)` mais `x` non lu (pas de `scanf`).
- `if (X < A)` lit `student_absent` via `scanf` ? L'étudiant saisit le nombre d'absents ?? Contresens.
- Condition arrêt `if (A == N)` dans la boucle.

**NOTE FINALE : 07 / 20**

**Feedback :**

- **Appréciation globale : Insuffisant.**
-

**Analyse :****Algorithmique :**

- Lectures correctes.
- Boucle `for` sur N avec lecture interne correcte.
- Logique interne correcte (compteurs P, D).
- Affichage final calculé.
- Condition finale `if (T >= S)` compare le nombre total d'étudiants au seuil ? Confusion. Devrait être P (absents selon sa logique P++ si `x < A`).

**Notation :**

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Initialisation	3 / 3	OK.
Condition boucle	2 / 4	Ok mais pas d'arrêt sur seuil.
Logique prés./abs.	4 / 4	OK.
Compteurs	3 / 3	OK.
Affichages inter.	2 / 2	OK.
Affichage final	0 / 1	Logique finale fausse.

**NOTE FINALE : 17 / 20****Feedback :**

- **Appréciation globale : Très Bon.**
-

**Analyse :**

**Algorithmique :**

- Utilise `write` au lieu de `printf`. Pseudo-code.
- Variables très longues explicites (`total_number...`) mais utilisées sans déclaration ou lecture adéquate dans une boucle.
- Pas de boucle explicite autour de la logique principale (seulement des `if` séquentiels).
- `if (x == N)` à la fin ?

**NOTE FINALE : 05 / 20**

**Feedback :**

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

**Analyse :**

**Algorithmique :**

- Lectures.
- Logic `if (n < A)` hors boucle.
- Boucle `while (A == N && A == S)` ? Condition quasi-impossible.
- Contenu incohérent.

**NOTE FINALE : 03 / 20**

**Feedback :**

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

Analyse :

Algorithmique :

- Initialisation `int i` manquante avant `while`.
- Boucle `while (i <= N || B == S)`. Condition arrêt fausse (`||` continue si seuil atteint).
- Réinitialisation `int B=0; int P=0; DANS la boucle !` Les compteurs sont remis à zéro à chaque tour.
- Logique d'incrémentation en fin de boucle. `i` n'est jamais incrémenté (boucle infinie).

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Initialisation	0 / 3	Compteurs remis à 0 dans la boucle. <code>i</code> non init.
Condition boucle	0 / 4	Boucle infinie ( <code>i</code> ne change pas).
Logique prés./abs.	4 / 4	OK.
Compteurs	0 / 3	Remis à zéro.
Affichage final	0 / 1	-

NOTE FINALE : 07 / 20

Feedback :

- **Appréciation globale : Insuffisant.** Erreur sur la portée des variables.