

Analyse :

Algorithmique :

- Lectures correctes.

Boucle `for (int i = 0; i <= N || i == S; i++)` :

- `i == S` dans une condition de continuation est dangereux. Probablement voulu dire `... && Absent < S`.
 - `i` redéclaré dans la boucle (déjà utilisé pour le numéro étudiant, masquage possible si `i` était externe, mais ici `i` est le compteur).
 - Comparaison `if (X < A)` correcte.
- Incrémentation : `Absent += 1, Present += 1` correcte.
- Attention : `Absent` et `Present` **non initialisés** ! Ils contiennent des valeurs aléatoires. `Absent += 1` donnera n'importe quoi.
 - Affichage final utilise une variable `Session` non initialisée (`if (Session >= S)`). Devrait être `Absent`.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initialisation	0 / 3	Variables d'accumulation non initialisées.
Condition boucle	2 / 4	Condition d'arrêt incorrecte (<code>i == S</code> au lieu de tester le compteur d'absents).
Logique prés./abs.	4 / 4	Correcte.
Compteurs	2 / 3	Logique d'incrémentation bonne, mais sur variables poubelles.
Affichages inter.	2 / 2	Présents.
Affichage final	0 / 1	Utilise une variable inconnue <code>Session</code> .

NOTE FINALE : 13 / 20

Feedback :

- Points forts** : Structure lisible, logique conditionnelle acquise.
- Points faibles** : **Initialisation des variables** ! C'est une erreur critique en C. Confusion sur les conditions d'arrêt.

- **Appréciation globale : Moyen.**
-