Rapport de Correction - Examen ASD S1 2025-2026

Fichier Source : `doc20260115201344.pdf` **Nombre de Copies** : 24

COPY NUMBER: 1

Analyse :

- **Contraintes** : Respectées (une seule boucle, stdio.h, pas de tableau).

  Algorithmique :
  - Erreur syntaxe déclaration : `const N;` au lieu de `int N;` ou `const int N = ...;`.
  - Boucle `for (i = 0; i <= x; i++)` : La condition d'arrêt dépend de `x` (variable non initialisée ou saisie dans la boucle), et non de `N` ou `S`. Logique de boucle incorrecte.
  - Lecture de `x` à l'intérieur de la boucle deux fois (printf puis scanf).
  - Le `printf` final utilise une variable `number of present and absent students` qui n'existe pas (pseudo-code non valide en C).
  - Condition finale `if (number of Absent student > S)` utilise aussi du pseudo-code.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	2 / 3	`scanf` présents mais déclaration `const N` invalide.
Initalisation	1 / 3	Variables non initialisées correctement (notamment compteurs).
Condition boucle	0 / 4	`i <= x` est insensé ici.
Logique prés./abs.	2 / 4	`if (x < A)` correct, mais dans une boucle mal structurée.
Compteurs	0 / 3	Aucun compteur incrémenté.
Affichages inter.	1 / 2	Messages présents mais pas de compteurs affichés.
Affichage final	0 / 1	Utilisation de pseudo-code invalide (`number of...`).

NOTE FINALE : 06 / 20

Feedback :

- **Points forts** : Structure globale du programme (main, includes, structure if/else).
- **Points faibles** : Confusion entre code C et pseudo-code (noms de variables). Logique de la boucle `for` à revoir complètement.

- **Appréciation globale** : **Fragile**. Les bases de la syntaxe sont à consolider. Attention à ne pas inventer de noms de variables avec des espaces.

COPY NUMBER: 2

Analyse :

- **Contraintes** : Respectées.

  Algorithmique :
  - Déclarations correctes.
  - Boucle `for` : Condition `i <= N || i <= S` est étrange (`||` au lieu de `&&` probablement, et comparaison `i <= S` incorrecte car `S` est un seuil d'absents, pas d'itérations). Indexation `i` commence à 1.
  - Lecture de `X` : Fait **avant** la boucle (donc une seule fois pour tous ?). Devrait être *dans* la boucle.
  - Prise de décision : Affiche "absence/present" mais n'incrémente pas de compteurs explicites pour `absent` ou `present`.
  - Condition finale : Utilise `M` (non initialisé) comparé à `S`. `M` semble être utilisé comme compteur mais n'est jamais incrémenté.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initalisation	1 / 3	`M` non initialisé.
Condition boucle	1 / 4	Utilisation de `||` rend la boucle potentiellement infinie ou incorrecte. `i <= S` n'a pas de sens ici.
Logique prés./abs.	2 / 4	Test `X < A` correct, mais `X` n'est pas lu à chaque itération.
Compteurs	0 / 3	Pas de mise à jour des compteurs.
Affichages inter.	1 / 2	Feedback textuel présent, mais pas de données chiffrées.
Affichage final	0 / 1	Basé sur une variable non initialisée `M`.

NOTE FINALE : 08 / 20

Feedback :

- **Points forts** : Saisie des paramètres correcte.

- **Points faibles** : Erreur logique majeure : `scanf` hors de la boucle. Gestion des compteurs inexistante. Confusion sur la condition d'arrêt (OU logique vs ET logique).
- **Appréciation globale** : **Insuffisant**. La logique répétitive n'est pas acquise.

COPY NUMBER: 3

Analyse :

- **Contraintes** : Manque `#include <stdio.h>`.
  Algorithmique :
  - Pas de boucle ! Le code s'exécute séquentiellement une seule fois (lit `N, A, S`, puis teste une fois).
  - Commentaire `// read the number of attended sessions x;` mais pas de `scanf` pour `X` avant son utilisation.
  - Logique conditionnelle dupliquée.
  - Pas de compteurs, pas de gestion de `N` étudiants.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initalisation	1 / 3	Variables déclarées mais `X` utilisé sans lecture (valeur indéterminée).
Condition boucle	0 / 4	**Aucune boucle présente.**
Logique prés./abs.	1 / 4	Structure `if` présente mais sur une variable non initialisée.
Compteurs	0 / 3	Absents.
Affichages inter.	0 / 2	Inexistants.
Affichage final	0 / 1	Affiche toujours "Session Valid".

NOTE FINALE : 05 / 20

Feedback :

- **Points forts** : Sait lire des entiers.
- **Points faibles** : L'énoncé demandait une simulation pour **N** étudiants. Absence totale de structure itérative. Code incomplet.
- **Appréciation globale** : **Très Insuffisant**. Hors sujet sur l'aspect algorithmique (boucles).

COPY NUMBER: 4

Analyse :

- **Contraintes** : Manque `#include <stdio.h>`.
  Algorithmique :
  - Lecture N, A, S correcte.
    Boucle `for (i = 0; i < N; i++)` correcte pour itérer N fois. **MAIS** ne gère pas l'arrêt anticipé si `B >= S`.
    - *Note*: Il y a un commentaire `// break // S`, l'étudiant savait qu'il fallait arrêter mais ne l'a pas codé.
  - `scanf("%d", &x)` dans la boucle : Correct.
  - Compteurs `B` (absents ?) et `P` (présents) incrémentés : Correct. **Cependant**, `B` et `P` ne sont **pas initialisés** à 0 ! (Comportement indéfini).
  - Logique `if (x < A)` correcte.
  - Affichage final cohérent.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initalisation	1 / 3	`B` et `P` non initialisés à 0.
Condition boucle	2 / 4	Boucle sur N correcte, mais condition d'arrêt sur S manquante dans le `for`.
Logique prés./abs.	4 / 4	Correcte.
Compteurs	3 / 3	Incrémentation correcte (malgré le défaut d'init).
Affichages inter.	1 / 2	Affiche le statut mais pas les compteurs courants à chaque étape.
Affichage final	1 / 1	Correct.

NOTE FINALE : 15 / 20

Feedback :

- **Points forts** : Bonne structure de boucle, lecture des entrées au bon endroit.
- **Points faibles** : **Oubli critique de l'initialisation** (`int R, P = 0;`). Condition d'arrêt sur le seuil d'absence manquante (le commentaire ne suffit pas).

- **Appréciation globale** : **Moyen / Bon**. Algorithme quasi-fonctionnel, erreurs d'inattention coûteuses.

COPY NUMBER: 5

Analyse :

- **Contraintes** : Manque `#include`.

  Algorithmique :
  - Lecture de N.
  - Boucle `for (i = 1; i <= N; i++)` : Correcte pour N itérations.
  - Lecture `scanf("%d %d", &A, &x)` **DANS** la boucle : Erreur majeure. `A` (seuil) ne doit être lu qu'une seule fois au début. Ici l'utilisateur doit le resaisir à chaque étudiant.
  - Lecture de `S` **APRÈS** la boucle : Trop tard. Impossible d'arrêter la simulation si le seuil est atteint pendant.
  - Logique `if (x < A)` correcte.
  - Pas de compteurs. Affiche juste le statut textuel.
  - Condition finale `if (N >= S)` : Compare le nombre total d'étudiants au seuil ? Sens douteux.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	1 / 3	Erreur de placement (A dans la boucle, S après).
Initalisation	3 / 3	Variables déclarées (`int` par défaut 0 non garanti mais ici pas de compteurs utilisés).
Condition boucle	2 / 4	Boucle sur N correcte, arrêt sur S impossible.
Logique prés./abs.	4 / 4	Comparaison correcte.
Compteurs	0 / 3	Inexistants.
Affichages inter.	1 / 2	Texte uniquement.
Affichage final	0 / 1	Logique incorrecte (`N >= S`).

NOTE FINALE : 11 / 20

Feedback :

- **Points forts** : Syntaxe de base correcte.
- **Points faibles** : Mauvaise compréhension de l'ordre des instructions (saisie des paramètres vs traitement). Gestion de l'arrêt anticipé manquante.
- **Appréciation globale** : **Moyen**.

COPY NUMBER: 6

Analyse :

Algorithmique :

- Lectures correctes.

  Boucle `for (i = 1; x <= N || x <= S; i++)` :

  - Condition conditionnée par `x` (variable de saisie courante ?) au lieu de compteurs.
  - Mélange `||` et variables non pertinentes.
- Lecture de `x` avant la boucle ? Non, il y a un scanf avant, mais pas **dans** la boucle (boucle infinie avec la même valeur de x, ou arrêt immédiat).
- Utilisation de `absent` dans le `if` final, mais variable non déclarée ni calculée.
- `return 0` **dans** la boucle, ce qui arrête le programme dès la première itération.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initalisation	2 / 3	Correct.
Condition boucle	0 / 4	Condition illogique et `scanf` manquant dans la boucle (bloqué ou infini).
Logique prés./abs.	2 / 4	Test présent, mais ne s'exécute pas correctement (pas de scanf interne).
Compteurs	0 / 3	Non implémentés.
Affichages inter.	1 / 2	Texte seul.
Affichage final	0 / 1	Variable `absent` non définie. Retour prématuré.

NOTE FINALE : 08 / 20

Feedback :

- **Points forts** : Début correct.
- **Points faibles** : Grosses erreurs de logique de contrôle (boucle mal formée, return prématuré, variable non déclarée). Le programme s'arrête immédiatement.
- **Appréciation globale** : **Fragile**.

COPY NUMBER: 7

Analyse :

Algorithmique :

- Lecture correcte.
- Boucle `for (int i = 1; i <= N; i++)`. Pas de condition d'arrêt sur S.
  Test `if (i < A)` : Compare le **numéro de l'étudiant** (`i`) au seuil d'absence (`A`) !
  - Erreur fondamentale. Il fallait lire une variable `X` (heures) et la comparer à `A`.
- Pas de lecture de la présence (`scanf`) dans la boucle.
- Conditions finales confuses (`if (i == N || N == S)`).

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initalisation	3 / 3	OK.
Condition boucle	2 / 4	Itère N fois, mais ne lit rien.
Logique prés./abs.	0 / 4	Compare l'indice de boucle au lieu d'une saisie.
Compteurs	0 / 3	Inexistants.
Affichages inter.	1 / 2	Texte basé sur une logique fausse.
Affichage final	0 / 1	Incohérent.

NOTE FINALE : 09 / 20

Feedback :

- **Points forts** : Code propre visuellement.
- **Points faibles** : Confusion totale sur le fonctionnement : on ne simule pas la présence en comparant le numéro de l'étudiant au seuil. Il manque la saisie des données pour chaque étudiant.
- **Appréciation globale** : **Insuffisant**. Le concept de traitement de données est manqué.

COPY NUMBER: 8

Analyse :

Algorithmique :

- Lecture groupée `scanf("%d %d %d", ...)` correcte.
- Lecture de `X` **avant** la boucle. Ne sera pas mis à jour.
- Boucle `for (i = 1; i <= N; i++)`.
- Corps de boucle : Affiche "absent/present" selon la valeur initiale de `X`.
- Affiche `PresentS` et `AbsentS` mais ces variables ne sont **jamais initialisées ni incrémentées**. Elles contiennent des valeurs poubelles.
- Test d'arrêt : `if (i == N || AbsentS == S)`. Comme `AbsentS` est poubelle, comportement imprévisible.
- `return 0` **dans** la boucle. S'arrête après la 1ère itération.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initalisation	0 / 3	Variables compteurs non initialisées.
Condition boucle	1 / 4	Boucle interrompue par un `return` prématuré.
Logique prés./abs.	2 / 4	Comparaison correcte mais sur une donnée fixe.
Compteurs	0 / 3	Aucune mise à jour.
Affichages inter.	0 / 2	Affiche des valeurs poubelles.
Affichage final	0 / 1	Code jamais atteint ou incorrect.

NOTE FINALE : 06 / 20

Feedback :

- **Points forts** : Syntaxe `scanf` concise.
- **Points faibles** : Variables non initialisées (grave en C). `scanf` hors boucle. `return` qui tue la boucle. Usage de variables non calculées.
- **Appréciation globale** : **Très Fragile**. Des lacunes importantes en algorithmie.

COPY NUMBER: 9

Analyse :

Algorithmique :

- Lectures correctes.
- Boucle `for (i = 0; i < n; i++)` : Correcte.
- `scanf("%d", &x)` dans la boucle : Correct.
- Logique : `if (a > x)` (Absent, `sum` non incrémenté ? Ah si `sum = sum + 1`).
- `else if (x >= a)` (Présent, `sum = sum` inutile).
- Utilise `sum` pour compter les absents. Pas de compteur pour présents affiché distinctement.
- Tests d'arrêt dans la boucle : `if (i == n || n == s)` -> Condition `n == s` est constante et ne dépend pas de l'avancement. Devait être `sum == s`.
- Pas de `break` explicite demandé mais condition d'arrêt mal formulée (affiche juste "stop in" sans arrêter).

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initalisation	3 / 3	`sum` initialisé.
Condition boucle	2 / 4	Boucle sur N ok, mais l'arrêt sur seuil n'est pas effectif (juste un print).
Logique prés./abs.	4 / 4	Correcte.
Compteurs	2 / 3	Compte les absents (`sum`), mais oublie de gérer/afficher les présents distinctement.
Affichages inter.	1 / 2	Affiche le cumul d'absents.
Affichage final	1 / 1	Logique `a > s` ? Non, devrait comparer `sum` et `s`. Confus.

NOTE FINALE : 16 / 20

Feedback :

- **Points forts** : Code fonctionnel sur la partie itérative de base.
- **Points faibles** : Confusion entre les variables (`a > s` à la fin au lieu de `sum > s`). L'arrêt sur seuil n'arrête pas vraiment la boucle (`break` interdit, il fallait mettre la condition dans le `for`).

- **Appréciation globale** : **Bon**. L'étudiant a compris le principe général.

COPY NUMBER: 10

Analyse :

Algorithmique :

- Lecture groupée OK.
- Boucle `while (St != N || Abs != S)` : L'opérateur devrait être `&&` (Tant que pas fini ET pas seuil atteint). Avec `||`, la boucle continue tant que l'une des conditions est vraie (risque de dépassement).
- Incrémentation `St++`, Lecture `x`, IF/ELSE pour `Abs`/`Pres`. Tout est correct.
- Affichages intermédiaires complets.
- Affichage final complet : `printf("Students: %d...", St...)`.
- Condition finale : `if (Abs < S)` Correct.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initalisation	3 / 3	Correcte.
Condition boucle	2 / 4	Erreur de logique booléenne : `||` au lieu de `&&` pour une boucle "Tant que".
Logique prés./abs.	4 / 4	Correcte.
Compteurs	3 / 3	Corrects.
Affichages inter.	2 / 2	Complets.
Affichage final	1 / 1	Correct.

NOTE FINALE : 18 / 20

Feedback :

- **Points forts** : Code très propre, logique claire, affichages conformes.
- **Points faibles** : Attention aux lois de Morgan (`while` continue tant que condition VRAIE -> `(St < N && Abs < S)`).
- **Appréciation globale** : **Très Bon**. Excellente copie.

COPY NUMBER: 11

Analyse :

Algorithmique :

- Lectures correctes.
- Utilise `x` avant de le lire (dans le premier `if` hors boucle).
- Boucle `while (i <= N)` vide (`// ...`). Code non terminé.
- Affichage final basé sur `x`.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initalisation	1 / 3	`x` non initialisé.
Condition boucle	0 / 4	Boucle vide.
Logique prés./abs.	1 / 4	Hors boucle et prématurée.
Compteurs	0 / 3	-
Affichages inter.	0 / 2	-
Affichage final	0 / 1	-

NOTE FINALE : 05 / 20

Feedback :

- **Points forts** : Début correct.
- **Points faibles** : Copie non finie. La logique principale est absente.
- **Appréciation globale** : **Insuffisant**.

COPY NUMBER: 12

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.
Initalisation	0 / 3	Variables d'accumulation non initialisées.
Condition boucle	2 / 4	Condition d'arrêt incorrecte (`i == S` 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 `Session`.

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**.

COPY NUMBER: 13

Analyse :

Algorithmique :

- Erreurs syntaxe `scanf`: `&total`, `&min_num`... Les variables déclarées sont `N, A, S`. Noms incohérents.
- Boucle `while (if (X < A))` : Syntaxe invalide. Pas de `if` dans une condition `while` de cette manière.
- Affectations incorrectes : `X = absent`. On écrase la saisie ? `absent` et `Present` semblent être utilisés comme des variables mais non déclarées.
- Arguments du `scanf` manquants (pour `S` ?).
- Code très confus.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	1 / 3	Noms de variables incohérents avec la déclaration.
Initalisation	0 / 3	-
Condition boucle	0 / 4	Syntaxe invalide.
Logique prés./abs.	0 / 4	Incompréhensible.
Compteurs	0 / 3	-
Affichages inter.	0 / 2	-
Affichage final	0 / 1	-

NOTE FINALE : 01 / 20

Feedback :

- **Appréciation globale** : **Très Insuffisant**. Problèmes majeurs de syntaxe et de cohérence. Revoir les bases impérativement.

COPY NUMBER: 14

Analyse :

Algorithmique :

- Lectures correctes.
- Boucle `for` avec condition `i <= N || i == S`. Arrêt sur `S` (seuil absences) incorrect car comparé à `i` (compteur tour).
- Saisie `scanf("%d", &X)`.
- Incrémentation : `Absent += i`. **Erreur** : Ajoute le numéro de l'étudiant (`i`) au lieu de 1 !
- `Absent` et `Present` non initialisés.
- Affichage final utilise `Session` (non déclaré).

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initalisation	0 / 3	Non faite.
Condition boucle	1 / 4	Condition d'arrêt sur seuil incorrecte.
Logique prés./abs.	4 / 4	Correcte.
Compteurs	0 / 3	Ajoute `i` au lieu de 1.
Affichages inter.	2 / 2	Présents.
Affichage final	0 / 1	Variable inexistante.

NOTE FINALE : 10 / 20

Feedback :

- **Points forts** : Code structuré qui ressemble à une solution.
- **Points faibles** : Incrémentation fausse (`+= i`), défaut d'initialisation, variables fantômes (`Session`).
- **Appréciation globale** : **Moyen -**.

COPY NUMBER: 15

Analyse :

Algorithmique :

- Initialisation correcte (`P=0, T=0`).
- Boucle `while (i <= N && T < S)` : **Excellente condition**. Gère les deux cas d'arrêt avec un ET logique.
- Saisie et tests corrects.
- Incrémentation correcte.
- Affichages complets.
- Décision finale correcte.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initalisation	3 / 3	Correct.
Condition boucle	4 / 4	Parfaite (&&).
Logique prés./abs.	4 / 4	Correct.
Compteurs	3 / 3	Correct.
Affichages inter.	2 / 2	Correct.
Affichage final	1 / 1	Correct.

NOTE FINALE : 20 / 20

Feedback :

- **Points forts** : Code parfait. Respect total des contraintes et de la logique. Bravo.
- **Appréciation globale** : **Très Bon**.

COPY NUMBER: 16

Analyse :

Algorithmique :

- Déclaration variables string/char `absent`, `present` inutiles ou mal utilisées.
- Condition `if (x < A)` **avant** la boucle et avant lecture cohérente.
- Boucle `while (i <= N)`. Corps de boucle : `na = N - na`. Calculs mathématiques étranges au lieu d'incrémentation simple.
- Saisie `scanf` pour tous les résultats à la fin ?? ("enter student number..."). L'étudiant redemande les résultats à l'utilisateur au lieu de les calculer.
- Pseudo-code dans les conditions (`if all students are processed...`).

Notation :

Critère	Points	Commentaire
Lecture N, A, S	2 / 3	Partiel.
Initalisation	0 / 3	-
Condition boucle	1 / 4	Boucle sur N uniquement.
Logique prés./abs.	1 / 4	Confuse.
Compteurs	0 / 3	Formules incorrectes.
Affichages inter.	0 / 2	-
Affichage final	0 / 1	Demande à l'utilisateur de saisir le résultat !

NOTE FINALE : 04 / 20

Feedback :

- **Points faibles** : Utilisation de pseudo-code. Ne calcule pas les résultats mais demande à l'utilisateur de les entrer à la fin.
- **Appréciation globale** : **Très Insuffisant**.

COPY NUMBER: 17

Analyse :

Algorithmique :

- Boucle `while (i < N || i == S)`. Condition d'arrêt incorrecte et mauvaise variable (`i` vs compteur d'absents).
- Utilisation de `X` sans lecture préalable dans la boucle (pas de scanf).
- `return 0` dans la boucle.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initalisation	1 / 3	Variables par défaut.
Condition boucle	1 / 4	Incorrecte.
Logique prés./abs.	1 / 4	Pas de lecture de X.
Compteurs	0 / 3	-
Affichages inter.	0 / 2	-
Affichage final	0 / 1	-

NOTE FINALE : 06 / 20

Feedback :

- **Appréciation globale** : **Insuffisant**. Programme qui ne fait rien (pas de lecture de données dans la boucle).

COPY NUMBER: 18

Analyse :

Algorithmique :

- Lectures correctes.
- Boucle `while (i <= N)`.
- Condition d'arrêt `if (i == S)` : Compare indice boucle au seuil d'absence (Faux). De plus, fait un `return 0` brutal. (Interdiction de `break`/`continue` contournée par `return` ou `if` structurant? `return` quitte le prog, donc plus d'affichage final).
- Calculs : Compare `absent_students > A` pour annulation ? C'est `> S`. Utilise des variables non déclarées/init (`absent_students`).

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initalisation	1 / 3	Variables manquantes.
Condition boucle	2 / 4	Boucle sur N ok, arrêt seuil mal géré.
Logique prés./abs.	3 / 4	Correcte.
Compteurs	0 / 3	Variables non déclarées.
Affichages inter.	1 / 2	Présent.
Affichage final	0 / 1	Incorrect.

NOTE FINALE : 10 / 20

Feedback :

- **Appréciation globale** : **Moyen -**. Logique présente mais implémentation défaillante (variables, conditions).

COPY NUMBER: 19

Analyse :

Algorithmique :

- Initialisation `P=0, a=0`.
- Boucle `for (i = 1; i <= N || a > S; i++)` : Condition `||` (OU) signifie que la boucle continue SI on dépasse le seuil !! C'est l'inverse (`&& a < S`).
- Lectures et tests corrects.
- Incrémentation correcte.
- Affichage final correct.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initalisation	3 / 3	Correct.
Condition boucle	2 / 4	Erreur de logique booléenne (`||` prolonge la boucle au lieu d'arrêter).
Logique prés./abs.	4 / 4	Correct.
Compteurs	3 / 3	Correct.
Affichages inter.	2 / 2	Correct.
Affichage final	1 / 1	Correct.

NOTE FINALE : 18 / 20

Feedback :

- **Points forts** : Très bon code.
- **Points faibles** : Petite erreur de logique sur la condition d'arrêt (`||` vs `&&`).
- **Appréciation globale** : **Très Bon**.

COPY NUMBER: 20

Analyse :

Algorithmique :

- Initialisation correcte.
- Boucle `for` OK. Pas d'arrêt prématuré explicite dans la condition du for (sauf `i<=N`), mais un `if (absent == S)` avec un message. Mais pas de `break` (interdit) ni modif de `i`. Donc la boucle continue.
- `return 0` dans la boucle après affichages finaux ? Le `return` est à la fin du bloc if/else final, qui est DANS la boucle ? Non, indentation suggère hors boucle.
- Logique propre.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initalisation	3 / 3	Correct.
Condition boucle	2 / 4	Ne gère pas l'arrêt effectif sur seuil (affiche juste le message).
Logique prés./abs.	4 / 4	Correct.
Compteurs	3 / 3	Correct.
Affichages inter.	2 / 2	Correct.
Affichage final	1 / 1	Correct.

NOTE FINALE : 18 / 20

Feedback :

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

COPY NUMBER: 21

Analyse :

Algorithmique :

- Boucle `for (int i = 0, i <= X, i++)` : Condition sur `X` (non initialisé) !
- Pseudo-code dans la condition `if (all N student are processed...)`.
- Code très incomplet.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	0 / 3	Absente.
Initalisation	0 / 3	-
Condition boucle	0 / 4	Incorrecte.
Logique prés./abs.	1 / 4	Sommaire.
Compteurs	0 / 3	-
Affichages inter.	0 / 2	-
Affichage final	0 / 1	-

NOTE FINALE : 01 / 20

Feedback :

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

COPY NUMBER: 22

Analyse :

Algorithmique :

- Lectures correctes.
- Utilise `X` sans le lire (scanf absent).
- Pas de compteurs.
- Condition finale `if (A >= N)` incohérente.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initalisation	1 / 3	Incomplète.
Condition boucle	1 / 4	Basique.
Logique prés./abs.	1 / 4	Pas de saisie de données.
Compteurs	0 / 3	-
Affichages inter.	0 / 2	-
Affichage final	0 / 1	Incorrect.

NOTE FINALE : 06 / 20

Feedback :

- **Appréciation globale** : **Insuffisant**. Pas fonctionnel.

COPY NUMBER: 23

Analyse :

Algorithmique :

- Boucle `for (i = 1; i <= N || i == S; i++)` : Condition d'arrêt sur `i == S` (indice vs seuil).
- Calculs `absent students = N - present students` mais `present students` non initialisé.
- Logique circulaire.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Initalisation	0 / 3	Manquante.
Condition boucle	2 / 4	Erreur logique indice/seuil.
Logique prés./abs.	2 / 4	OK pour la condition.
Compteurs	0 / 3	Calculs faux.
Affichages inter.	1 / 2	OK.
Affichage final	0 / 1	OK.

NOTE FINALE : 08 / 20

Feedback :

- **Appréciation globale** : **Fragile**.

COPY NUMBER: 24

Analyse :

Algorithmique :

- Code fragmentaire.
- `if (x < 0)` : Condition étrange (devrait être `A`).
- `scanf("%d", absents_student)` : Utilise scanf en écriture ??
- Boucle `for (i = N)` syntaxiquement fausse.

NOTE FINALE : 02 / 20

Feedback :

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