

## 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/présent" 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.
Initialisation	0 / 3	Variables compteurs non initialisées.
Condition boucle	1 / 4	Boucle interrompue par un <code>return</code> 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.