

Input reading: 3pts | Initialization: 3pts | Loop condition: 4pts | Counters logic: 4pts | Stop conditions: 3 | Final output: 3

include <stdio.h>

int main() {

int N, A, S, ⁱ⁼¹ i = 1, X, Present S, Absent S, Processed S;

scanf("%d %d %d", &N, &A, &S);

printf("Enter the number of attended sessions");

scanf("%d", &X);

for (i = 1, i <= N, i++) {

. if (X < A) {

printf("the student is absent");

} else {

printf("the student is present :)"); }

printf("%d is the number of present students", Present S);

printf("%d is the number of absent students", Absent S);

. if (i == N || Absent S == S) {

printf("stop the simulation !!");

} else

~~printf("Enter the number of present students");~~

scanf("%d", &Present S);

printf("Session valid");

scanf("%d", &Absent S);

printf("Session cancelled");

return 0; }

}

Copy 8

```
#include <stdio.h>
int main()
{
    int N, A, S, i = 1, X, PresentS, AbsentS, ProcessedS;
    scanf("%d %d %d", &N, &A, &S);
    printf("Enter the number of attended sessions");
    scanf("%d", &X);
    for (i = 1; i <= N; i++)
    {
        if (X < A)
        {
            printf("the student is absent");
        }

        else
        {
            printf("the student is present");
        }

        printf("%d is the number of present students", PresentS);
        printf("%d is the number of absent students", AbsentS);
        if (i == N || AbsentS == S)
        {
            printf("stop the simulation !!");
        }

        else
        {
            // printf("Enter the number of present students");
            // scanf("%d", &PresentS);
            printf("Session valid");
            // scanf("%d", &AbsentS);
            printf("Session cancelled");
        }

        return 0;
    }
}
```

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.