

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

```

int main() {
    int N, A, S, x, i, B, P;
    printf("Enter a total number of registered students");
    scanf("%d", &N);
    printf("Enter the minimum attendance required");
    scanf("%d", &A);
    printf("Enter absence threshold");
    scanf("%d", &S);
    printf("Enter the number of attended sessions");
    for (i = 0; i <= N; i++) {
        scanf("%d", &x);
        if (x < A) {
            printf("The student is absent");
            B += i;
        }
        else if (x >= A) {
            printf("The student is present");
            P += i;
        }
        break // 5
    }
    printf("present students: %d", P);
    printf("absent students: \n %d", B);
    if (B >= S) {
        printf("Session cancelled");
    }
    else if (B < S) {
        printf("session valid");
    }
    return 0;
}

```

Loop  
Condition

P: حاضر  
B: غائب

## Copy 4

---

```
int main()
{
    int N, A, S, x, i, B, P;
    printf("Enter a total number of registered students");
    scanf("%d", &N);
    printf("Enter the minimum attendance required");
    scanf("%d", &A);
    printf("Enter absence threshold");
    scanf("%d", &S);
    printf("Enter the number of attended sessions");
    for (i = 0; i <N; i++)
    {
        scanf("%d", &x);
        if (x <A)
        {
            printf("the student is absent");
            B++;
        }

        else
        {
            if (x>= A)
            {
                printf("the student is present");
                P++;
            }

        }

    }

    // break // S printf("present students: %d", P);
    printf("absent students: %d", B);
    if (B>= S)
    {
        printf("Session cancelled");
    }

    else
    {
        if (B <S)
        {
            printf("session valid");
        }

    }

    return 0;
}
```

## 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.
Initialisation	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 <code>for</code> .
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.
-