

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

```

#include <stdio.h>
int main() {
    int n, a, s, p;
    int H = 0; // H is a counter
    int P = 0; // P is a counter
    int i = 1
    int f = n // f is a counter
    printf("enter number of student");
    scanf("%d", &n);
    printf("enter number of minimum attendance required");
    scanf("%d", &a);
    printf("enter number of absence threshold");
    scanf("%d", &s);
    do {
        for (i; i <= n; i++) {
            printf("enter the number of the attended sessions for the %d student", i);
            scanf("%d", &n);
            if (n < a) {
                printf("the student number %d is absence", i);
                H++;
            }
            else {
                printf("the student is not absence %d it's number", i);
                P++;
            }
            printf("%d is the number of absences", H);
            printf("%d is the number of presence", P);
        }
        while (f != 0 || H < s);
        if (H > s) {
            printf("session cancelled");
        }
        else {
            printf("session valid");
            printf("%d absences", H);
            printf("%d presences", P);
        }
        return 0;
    }

```

## Copy 6

---

```
#include <stdio.h>
int main()
{
    int n, a, s, x, i;
    int H = 0;
    int P = 0;
    int f = n;
    // fis a counter printf("enter number of student");
    scanf("%d", &n);
    printf("enter number of minimum attendance required");
    scanf("%d", &a);
    printf("enter number of absence threshold");
    scanf("%d", &s);
    do
    {
        for (i = 1; i <= N; i++)
        {
            printf("enter thenumber of the attended sessions for the student %d", i);
            scanf("%d", &x);
            if (x <a)
            {
                printf("the student number %d is absence", i);
                H++;
            }

            else
            {
                printf("the student is not absence %d, it's number", i);
                P++;
            }

        }

    }

    while (f != 0 || H <S);
    if (H> S)
    {
        printf("session cancelled");
    }

    else
    {
        printf("session valid");
        printf("%d absences", H);
        printf("%d presences", P);
    }

    return 0;
```



Analyse :

- Algorithmique :**
- Boucle `do { for ... } while` complexe et inutile.
  - Boucle `for` sur `N` avec lecture interne correcte.
  - Le `do-while` extérieur dépend de `H < S` (seuil). C'est une façon originale de gérer l'arrêt, mais le `for` interne va quand même traiter tout le monde avant de vérifier le `while`. Donc arrêt tardif.
  - Logique interne correcte.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Initialisation	3 / 3	OK.
Condition boucle	2 / 4	Structure boucle imbriquée maladroite ne permettant pas l'arrêt immédiat au seuil.
Logique prés./abs.	4 / 4	OK.
Compteurs	3 / 3	OK.
Affichages inter.	2 / 2	OK.
Affichage final	1 / 1	OK.

NOTE FINALE : 18 / 20

Feedback :

- **Appréciation globale : Bon.** Structure un peu lourde mais fonctionne.
-