

```

#include <stdio.h>
int main () {
    int N, A, S;
    printf("Enter the number of registered students = \n");
    scanf("%d", &N);
    printf("Enter the minimum attendance required = \n");
    scanf("%d", &A);
    printf("Enter the absence threshold = \n");
    scanf("%d", &S);
    for (int i = 0; i <= N || i == S; i++) {
        if (int n < A) {
            i += S;
            printf("the student is absent");
            cont++;
        }
        else {
            printf("the student is present");
            cont++;
        }
    }
    printf("the absent students are = %d", cont);
    scanf("%d", &cont);
    printf("the present students are = %d", cont);
    printf("the number student is = ");
    else if (N > S) {
        printf("Session Cancelled");
    }
    else {
        printf("Session valid");
        return 0;
    }
}

```

Copy 1

```
#include <stdio.h>
int main()
{
    int N, A, S;
    printf("Enter the number of registered students: \n");
    scanf("%d", &N);
    printf("Enter the minimum attendance required: \n");
    scanf("%d", &A);
    printf("Enter the absence threshold = \n");
    scanf("%d", &S);
    for (int i = 0; i <= N || i == S; i++)
    {
        if (int x < A)
        {
            i = S;
            printf("the student is absent");
            cont++;
        }

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

    printf("the absent students are: %d", cont);
    printf("the present students are: %d", con);
    printf("the number student is: ");
    if (N > S)
    {
        printf("Session Cancelled");
    }

    else
    {
        printf("Session Valid");
    }

    return 0;
}
```

Analyse :**Algorithmique :**

- Lecture N, A, S correcte.

Boucle for (`int i = 0; i <= N || i == S; i++`) :

- Condition d'arrêt incorrecte (`i == S` au lieu de tester le nombre d'absents).
- Condition `i <= N` correcte mais indexation à 0 finit à N (soit N+1 itérations).
- Déclaration `if (int x < A)` à l'intérieur du `if` : Interdit ! `int x` déclare une nouvelle variable locale non initialisée qui masque tout. De plus, x n'a pas été lu (`scanf` manquant dans la boucle).
- `i = S` dans le `if` : Tentative de forcer la sortie de boucle ?
- `cont` non initialisé. `con` utilisé mais non déclaré.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Initialisation	0 / 3	Compteurs non initialisés/déclarés incorrectement.
Condition boucle	1 / 4	Condition d'arrêt fausse.
Logique prés./abs.	0 / 4	Erreur grave : redéclaration <code>int x</code> dans le <code>if</code> sans saisie.
Compteurs	0 / 3	Noms variables incohérents (<code>cont</code> vs <code>con</code>).
Affichages inter.	1 / 2	Texte sans valeurs.
Affichage final	0 / 1	-

NOTE FINALE : 05 / 20**Feedback :**

- Appréciation globale : Très Insuffisant.** Problèmes de portée de variables et de logique de base.
-

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=0;
    int A=0;
    int S=0;
    int absent=0;
    int present=0;
    int x=0;
    printf("%d %d %d", N, A, S);
    scanf("%d %d %d", &N, &A, &S);
    for (int i=1; i<=N; i++) {
        printf("enter x");
        scanf("%d", &x);
        if (x < A)
            printf("absent");
            absent++;
        else
            printf("the number of absent...");
            present++;
    }
    printf("ab = %d", S);
}

```

```

printf("number of ---- x");
scanf("%d", &x);
if (x < A)
    printf("absent");
else
    printf("present");
    present++;
printf("in");
printf("absent");
printf("present");
if (absent == S) {
    i = N;
}
printf("total n of student");
printf("present");
printf("absent");
if (absent == S)
    printf("Valid");
else
    printf("cancelled");
}

```

Copy 2

```
#include <stdio.h>
int main()
{
    int N = 0;
    int A = 0;
    int S = 0;
    int absent = 0;
    int present = 0;
    int x = 0;
    printf("N, A, S");
    scanf("%d %d %d", &N, &A, &S);
    for (int i = 1; i <= N; i++)
    {
        printf("entre x");
        scanf("%d", &x);
        if (x < A)
        {
            printf("absent");
            absent++;
        }

        else
        {
            printf("present");
            present++;
        }

        printf("number of ... x");
        scanf("%d", &x);
        if (x < A)
        {
            printf("absent");
            absent++;
        }

        else
        {
            printf("present");
            present++;
        }

        if (absent == S)
        {
            i = N;
        }
    }

    printf("total n of student");
```

```
printf("present");  
printf("absent");  
if (absent >= S)  
{  
    printf("Valid");  
}  
  
else  
{  
    printf("cancelled");  
}  
  
return 0;  
}
```

Analyse :**Algorithmique :**

- Syntaxe `printf(N, A, S)` incorrecte (pas de format). `scanf(N, A, S)` incorrect (pas de format ni &).
- Initialisation `int N=0` effacée par le `scanf` (si ça marchait).
- Boucle `for (i=1; i<=N)`.
- Lecture `x` deux fois ?? `scanf("%x")` (hexadécimal ??) puis `scanf("%d", &x)`.
- Condition d'arrêt `if (absent == S) i = N;` (Astuce pour sortir, fonctionnelle).

Notation :

Critère	Points	Commentaire
Lecture N, A, S	0 / 3	Syntaxe <code>scanf</code> totalement fausse (pas de %d).
Initialisation	3 / 3	OK.
Condition boucle	2 / 4	Arrêt simulé via <code>i=N</code> . Ok.
Logique prés./abs.	2 / 4	Double saisie confuse. <code>scanf("%x")</code> .
Compteurs	3 / 3	OK.
Affichages inter.	2 / 2	OK.
Affichage final	1 / 1	OK.

NOTE FINALE : 13 / 20**Feedback :**

- **Appréciation globale : Moyen.** La logique est là, mais la syntaxe des E/S est à revoir d'urgence (`scanf`).
-


```
#include <stdio.h>
```

```
int main() {
```

```
    int N;
```

```
    int S;
```

```
    int A;
```

```
    int i;
```

```
    int X;
```

```
    int P, B;
```

```
    printf("Enter total number of registered students: N");
```

```
    scanf("%d", &N);
```

```
    printf("Enter the minimum attendance required ");
```

```
    scanf("%d", &A);
```

```
    printf("Enter the absence threshold: ");
```

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

```
    for (i = 1; i <= N; i++)
```

```
    {
        scanf("%d", &X);
```

```
        if (X < A)
```

```
            count++
```

```
        } else {
```

```
            count++
```

```
        }
```

```
        printf("%d", i);
```

```
        printf("The total of present students is: %d", P);
```

```
        printf("The total of absent students is: %d", B);
```

```
    }
```

```
    printf("The total of presents is: \n");
```

```
    printf("The total of absents is: \n");
```

```
    if (B < P || B = P)
```

```
    {
        printf("Session valid");
```

```
    }
```

```
    else
```

```
    {
        printf("Session valid cancelled");
```

```
    }
```


Copy 3

```
#include <stdio.h>
int main()
{
    int N, S, A, i, X;
    int P, B;
    printf("Enter total number of registered students N");
    scanf("%d", &N);
    printf("Enter the minimum attendance required");
    scanf("%d", &A);
    printf("Enter the absence threshold");
    scanf("%d", &S);
    for (i = 1; i <= N || B == S; i++)
    {
        scanf("%d", &X);
        if (X < A)
        {
            count++;
        }

        else
        {
            count++;
        }

    }

    printf("%d", i);
    printf("the total of present students is: %d", P);
    printf("the total of absent students is: %d", B);
    if (B < P || B == P)
    {
        printf("Session Valid");
    }

    else
    {
        printf("Session Cancelled");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Lectures correctes.
- Boucle for : Condition $B == S$ étrange pour *continuer* ? Devrait être $B < S$.
- Compteur count utilisé mais non déclaré. P et B déclarés mais non initialisés (et utilisés ensuite).
- Confusion entre count, P, B.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Initialisation	0 / 3	Variables non initialisées.
Condition boucle	1 / 4	Incorrecte.
Logique prés./abs.	2 / 4	Structure if ok, mais variables inconnues.
Compteurs	0 / 3	count non déclaré.
Affichage final	0 / 1	-

NOTE FINALE : 06 / 20

Feedback :

- **Appréciation globale : Fragile.** Code non compilable.
-

```

#include <stdio.h>
int main() {
    int N, A, S, X, i, absent, present;
    printf("total number of registered students");
    scanf("%d", &N);
    printf("mini min attendance required");
    scanf("%d", &A);
    printf("absent threshold");
    scanf("%d", &S);
    printf("Enter the number of attended sessions.");
    scanf("%d", &X);
    if (X >= A) {
        printf("the student is present");
        for (i = 0, i <= N; i++) {
            printf("present student");
            present = i;
        }
    }
    else {
        printf("the student absent");
        absent = N - i;
        printf("%d", &absent);
    }
    if (i >= S) {
        printf("Session cancelled");
    }
    else {
        printf("Session Valid");
    }
    return 0;
}

```

Copy 4

```
#include <stdio.h>
int main()
{
    int N, A, S, X, i, absent, present;
    printf("total number of registered students");
    scanf("%d", &N);
    printf("minimum attendance required");
    scanf("%d", &A);
    printf("absence threshold");
    scanf("%d", &S);
    printf("Enter the number of attended sessions");
    scanf("%d", &X);
    if (X >= A)
    {
        printf("the student is present");
        for (i = 0; i <= N; i++)
        {
            printf("present student");
            present = i;
        }
    }

    else
    {
        printf("the student absent");
        absent = N - i;
        printf("%d", &absent);
    }

    if (i >= S)
    {
        printf("Session cancelled");
    }

    else
    {
        printf("Session Valid");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Lectures correctes.
- Lecture de X **avant** la boucle (une seule fois pour tous).
- Si $X \geq A$, boucle for imbriquée qui affiche "present" N fois de suite avec la même valeur ? Logique totalement fausse.
- Pas de traitement par étudiant.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Condition boucle	0 / 4	Logique aberrante (boucle dans le if).
Logique prés./abs.	1 / 4	Test sur valeur unique.
Compteurs	0 / 3	-
Affichages inter.	0 / 2	-
Affichage final	0 / 1	-

NOTE FINALE : 04 / 20

Feedback :

- **Appréciation globale : Très Insuffisant.** Algorithme non compris.
-

Input reading: 3 pts | 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, X;
    printf("Enter total number of registered Students");
    scanf("%d", &N);
    printf("Enter minimum attendance required");
    scanf("%d", &A);
    printf("Enter absence threshold");
    scanf("%d", &S);
    printf("Enter the number of attended Sessions");
    scanf("%d", &X);
    for (i=0; i<X; i++) {
        if (X<A)
            printf("the student is absent");
        else
            printf("the student is present");
    }
}

```

```

return 0;
}

```

Copy 5

```
#include <stdio.h>
int main()
{
    int N, A, S, X;
    printf("Enter total number of registered students");
    scanf("%d", &N);
    printf("Enter minimum attendance required");
    scanf("%d", &A);
    printf("Enter absence threshold");
    scanf("%d", &S);
    printf("Enter the number of attended sessions");
    scanf("%d", &X);
    for (int i = 0; i <X; i++)
    {
        if (X <A)
        {
            printf("the student is absent");
        }

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

    }

    return 0;
}
```


Analyse :

Algorithmique :

- Lectures correctes.
- Lecture de X **avant** la boucle.
- Boucle for (i=0; i<X; i++) : Itère X fois (nombre de séances) au lieu de N fois (nombre d'étudiants). Contresens complet.
- La boucle répète juste le statut de l'étudiant unique X fois.

NOTE FINALE : 05 / 20

Feedback :

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

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 the number of absences
    int P = 0; // P is the number of presences
    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 presences", 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 <code>N</code> , <code>A</code> , <code>S</code>	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.
-

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, x, j = 0, K = 0;
    printf("enter the number of registered students");
    scanf("%d", &N);
    printf("enter the minimum attendance required");
    scanf("%d", &A);
    printf("enter the absence threshold");
    scanf("%d", &S);
    for (i = 1; i <= n; i++) {
        printf("give me the number of sessions of student number %d", i);
        scanf("%d", &x);
        if (x < A) {
            printf("the student number %d is absent", i);
            j = j + 1;
        } else {
            printf("the student number %d is present", i);
            K = K + 1;
        }
    }
    printf("the number of present student is %d", K);
    printf("the number of absent student is %d", j);
    if (j >= S) {
        printf("session cancelled");
    } else {
        printf("session valid");
    }
    return 0;
}

```

Copy 7

```
#include <stdio.h>
int main()
{
    int N, A, S, x, j = 0, K = 0;
    printf("enter the number of registered students");
    scanf("%d", &N);
    printf("enter the minimum attendance required");
    scanf("%d", &A);
    printf("enter the absence threshold");
    scanf("%d", &S);
    for (int i = 0; i <= n; i++)
    {
        printf("give me the number of sessions of student number %d", i);
        scanf("%d", &x);
        if (x < A)
        {
            printf("the student number %d is absent", i);
            j = j + 1;
        }

        else
        {
            printf("the student number %d is present", i);
            K = K + 1;
        }

    }

    printf("the number of present student is %d", K);
    printf("the number of absent student is %d", j);
    if (j >= S)
    {
        printf("session cancelled");
    }

    else
    {
        printf("session valid");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Variable n déclarée mais utilise n non init dans for. Ah, n est le paramètre lu ? Non, scanf("%d", &N). Utilise N et n de façon interchangeable ? Code: i <= n. n n'est pas init.
- Utilise x (minuscule) dans scanf, X non déclaré ? Ou l'inverse.
- Utilise K et j pour les compteurs.
- Logic if (x < A) correcte.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	2 / 3	Confusion de noms de variables (n vs N).
Initialisation	1 / 3	n utilisé dans boucle non initialisé.
Condition boucle	1 / 4	Boucle risque de ne pas s'exécuter (n ?). Pas d'arrêt sur seuil.
Logique prés./abs.	3 / 4	OK.
Compteurs	3 / 3	OK.
Affichage final	1 / 1	OK.

NOTE FINALE : 11 / 20

Feedback :

- **Appréciation globale : Moyen -.** Attention à la casse et aux noms de variables.


```

include <stdio.h>
int main() {
    int N, A, S, X, Y, Z = 0, i = 1;
    printf("Enter the Number of registered students");
    scanf("%d", &N);
    printf("Enter The minimum attendance required");
    scanf("%d", &A);
    printf("Enter the Absence threshold");
    scanf("%d", &S);
    while (i <= N && Z != S) {
        printf("What is the Number of Attended session X");
        scanf("%d", &X);
        if (A > X) {
            printf("the student is absent");
            Z++;
        } else {
            printf("the student is present");
            Z++;
        }
        i++;
    }
    Y = N - Z;
    printf("total processed student = %d\n", N);
    printf("the Number of present students = %d\n", Y);
    printf("the Number of absent students = %d\n", Z);
    if (Z == S) {
        printf("Session cancelled");
        return 0;
    }
}

```

Copy 8

```
#include <stdio.h>
int main()
{
    int N, A, S, X, Y, Z = 0, i = 1;
    printf("Enter the number of registered students");
    scanf("%d", &N);
    printf("Enter the minimum attendance required");
    scanf("%d", &A);
    printf("Enter the absence threshold");
    scanf("%d", &S);
    while (i <= N && Z != S)
    {
        printf("What is the number of attended session X");
        scanf("%d", &X);
        if (A > X)
        {
            printf("the student is absent");
            Z++;
        }

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

    }

    Y = N - Z;
    printf("total processed student = %d", N);
    printf("the number of present students = %d", Y);
    printf("the number of absent students = %d", Z);
    if (Z == S)
    {
        printf("Session cancelled");
    }

    return 0;
}
```

Analyse :**Algorithmique :**

- Lecture OK.
- Boucle `while` bien construite (`i <= N && Z != S`).
- Lecture `X` dans la boucle.
- Logique `if (A > X) (Absent) OK`.
- Problème : `i++` est dans le `else` (présent) uniquement ! Si l'étudiant est absent, on n'incrémente pas `i`. Boucle potentiellement infinie si que des absents (mais arrêt sur `Z != S`). Par contre, si l'étudiant est absent, on reste sur le même `i` pour le suivant ? Logique étrange. On devrait incrémenter `i` tout le temps.

Notation :

Critère	Points	Commentaire
Lecture <code>N</code> , <code>A</code> , <code>S</code>	3 / 3	OK.
Condition boucle	4 / 4	Condition composée correcte.
Logique prés./abs.	2 / 4	Erreur : l'incréméntation de <code>i</code> est conditionnelle (seulement si présent).
Compteurs	3 / 3	OK.
Affichage final	1 / 1	OK.

NOTE FINALE : 13 / 20**Feedback :**

- **Appréciation globale : Moyen.** L'étudiant ne passe au suivant que s'il est présent ?
-

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;
    printf("total of processed students", S);
    if {
        N < A;
        printf("The student is absent");
    } else {
        printf("the student is present");
    }
    for {
        (N == 0; N <= A; N++)
        printf("Session valid");
    } if not {
        printf("Session cancelled");
    }
    return 0;
}

```

output :- total processed students

- present students
- absent students
- Session valid
- session cancelled

final status }

Copy 9

```
#include <stdio.h>
int main()
{
    int N, A, S, i;
    printf("total of processed students: ", N);
    if (X < A)
    {
        printf("the student is absent");
    }

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

    for (i = 0; i <= N; i++)
    {
        if (N == A || N++)
        {
            printf("Session valid");
        }

        if (not)
        {
            printf("Session cancelled");
        }

    }

    return 0;
}
```

Analyse :

Algorithmique :

- Condition `if (X < A)` placée **avant** la boucle et avant lecture de X.
- Boucle `for` sur N.
- Corps de boucle : `if (N == A || N++)`. N'a aucun sens. Ne lit rien.

NOTE FINALE : 02 / 20

Feedback :

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


```
#include <stdio.h>
```

```
int main() {
```

```
    int N, A, S;
```

```
    printf("Total number of registered students ");
```

```
    scanf("%d", &N);
```

```
    printf("Enter minimum attendance required");
```

```
    scanf("%d", &A);
```

```
    printf("Enter absence threshold");
```

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

```
    do { int i = 1, NS = 0, NA = 0;
```

```
        printf("Enter the number of attendance of student %d", i);
```

```
        scanf("%d", &X);
```

```
        if (X < A) {
```

```
            printf("Student %d is absent", i);
```

```
            NS = NS + 1;
```

```
        } else { printf("Student %d is present", i);
```

```
            NA = NA + 1;
```

```
            i++;
```

```
        while (i <= N || NS <= S) {
```

```
            printf("The number of present student is %d", NA);
```

```
            printf("The number of absent student is %d", NS);
```

```
            if (NS >= S) { printf("The session is cancelled");
```

```
            } else { printf("The session is valid"); }
```

```
        return 0; }
```

Copy 10

```
#include <stdio.h>
int main()
{
    int N, A, S, X, NS = 0, NA = 0;
    printf("total number of registered students ");
    scanf("%d", &N);
    printf("enter minimum attendance required ");
    scanf("%d", &A);
    printf("enter absence threshold ");
    scanf("%d", &S);
    do
    {
        int i = 1;
        printf("enter the number of attendance of student %d", i);
        scanf("%d", &X);
        if (X < A)
        {
            printf("student is absent");
            NS = NS + 1;
        }

        else
        {
            printf("student is present");
            NA = NA + 1;
        }

        i++;
    }

    while (i <= N || NS < S);
    printf("the number of present student is %d", NA);
    printf("the number of absent student is %d", NS);
    if (NS >= S)
    {
        printf("the session is cancelled");
    }

    else
    {
        printf("the session is valid");
    }

    return 0;
}
```


Analyse :

Algorithmique :

- Lecture OK.
- Boucle `do ... while`. Condition `NS < S`. Correct.
- Corps : lecture et tests OK.
- Affichages : OK.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Initialisation	3 / 3	OK.
Condition boucle	4 / 4	<code>do while</code> bien utilisé.
Logique prés./abs.	4 / 4	OK.
Compteurs	3 / 3	OK.
Affichages inter.	2 / 2	OK.
Affichage final	1 / 1	OK.

NOTE FINALE : 20 / 20

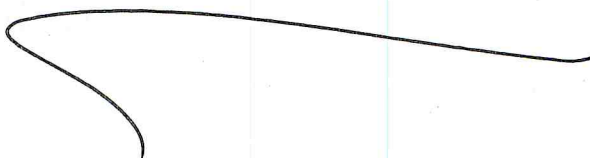
Feedback :

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

```

#include <stdio.h>
int main() {
    int N, A, S, Sum1=0, Sum2=0;
    scanf("%d", &N);
    scanf("%d", &A);
    scanf("%d", &S);
    for (int i=1; i<=N; i++) {
        if (S > N) {
            int x;
            scanf("%d", &x);
            if (x < A) {
                printf("absent student");
                Sum1 = Sum1 + Sum2;
            }
            else {
                printf("print student");
                Sum2 = Sum2 + Sum2;
            }
            printf("the number of absent students = %d", Sum1);
            printf("the number of print student = %d", Sum2);
            if (Sum1 > Sum2) {
                printf("Session cancelled");
            }
            else {
                printf("Session valid");
            }
        }
        return 0;
    }
}

```



Copy 11

```
#include <stdio.h>
int main()
{
    int N, A, S, Sum1 = 0, Sum2 = 0;
    scanf("%d", &N);
    scanf("%d", &A);
    scanf("%d", &S);
    for (int i = 1; i <= N || S > N; i++)
    {
        int x;
        scanf("%d", &x);
        if (x < A)
        {
            printf("absent student");
            Sum1 = Sum1 + Suma;
        }

        else
        {
            printf("present student");
            Sum2 = Sum2 + Suma;
        }

        printf("the number of absent students = %d", Sum1);
        printf("the number of present student = %d", Sum2);
        if (Sum1 > Sum2)
        {
            printf("session cancelled");
        }

        else
        {
            printf("session valid");
        }

    }

    return 0;
}
```

Analyse :

Algorithmique :

- Condition boucle for (i = 1; i <= N || S > N; i++). Le || S > N rend la condition toujours vraie si S > N (ce qui est possible). Boucle infinie.
- Sum1 = Sum1 + Suma. Suma n'est pas déclaré. Voulait dire 1 ? ou Sum1 ?
- Logique de comptage fausse.

NOTE FINALE : 07 / 20

Feedback :

- **Appréciation globale : Insuffisant.** Code non fonctionnel (variables inconnues).
-

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

```

#include <stdio.h>
#define N 100
#define A 5
#define S 10

int main() {
    int X, nbre-pr = 0, nbre-ab = 0, i = 0;
    while (i < N || nbre-ab < S)
    {
        printf("enter the number of attended sessions : ");
        scanf("%i", &X);
        if (X < A) {
            nbre-ab++;
        }
        else {nbre-pr++;}
        i++;
        printf("present students = %i\n", nbre-pr);
        printf("absent students = %i\n", nbre-ab);
        printf("step number %i\n", i);
    }
    printf("Total processed students = %i\n", i);
    printf("present students = %i\n", nbre-pr);
    printf("absent students = %i\n", nbre-ab);
    if (nbre-ab < S) { printf("Session valid"); }
    else { printf("Session cancelled"); }
    return 0; }

```

Copy 12

```
#include <stdio.h>
#define N 100 #define A 5 #define S 10 int main()
{
    int X, nbre_pr = 0, nbre_ab = 0, i = 0;
    while (i <N || nbre_ab <S)
    {
        printf("enter the number of attended sessions: ");
        scanf("%d", &X);
        if (X <A)
        {
            nbre_ab++;
        }

        else
        {
            nbre_pr++;
        }

        i++;
        printf("present students = %d", nbre_pr);
        printf("absent students = %d", nbre_ab);
        printf("step number %d", i);
    }

    printf("Total processed students = %d", i);
    printf("present students = %d", nbre_pr);
    printf("absent students = %d", nbre_ab);
    if (nbre_ab <3)
    {
        printf("session valid");
    }

    else
    {
        printf("session cancelled");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Utilisation de #define pour N, A, S. L'énoncé disait "Lire N, A, S". (-3 pts lecture).
- Boucle while correcte.
- Logique correcte.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	0 / 3	Utilisation de constantes au lieu de scanf. Hors sujet.
Initialisation	3 / 3	OK.
Condition boucle	4 / 4	OK.
Logique prés./abs.	4 / 4	OK.
Compteurs	3 / 3	OK.
Affichages inter.	2 / 2	OK.
Affichage final	1 / 1	OK.

NOTE FINALE : 17 / 20

Feedback :

- **Appréciation globale :** Très Bon code, mais attention à bien lire l'énoncé (entrées dynamiques demandées).
-

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

```

#include <stdio.h>
int main()
{
    int A;
    int B;
    int sum = 0;
    int S;
    int x;
    printf("أدخل رقمين ");
    scanf("%d", &A);
    printf("أدخل الرقم ");
    scanf("%d", &A);
    printf("أدخل قيمة ");
    scanf("%d", &S);
    while ((N > S) || (A != S))
    {
        printf("أدخل الرقم ");
        scanf("%d", &x);
        if (x < A) printf("نماذج");
        else printf("ناظر");
        sum = sum + x;
        sum = sum + x;
        printf("النتيجة = %d", sum);
    }

    return 0;
}

```


Copy 13

```
#include <stdio.h>
int main()
{
    int n, sum = 0;
    int S, total, A, x;
    printf("enter total number");
    scanf("%d", &n);
    printf("enter minimum attendance");
    scanf("%d", &A);
    printf("enter absence threshold");
    scanf("%d", &S);
    while (n> S || A == 5)
    {
        printf("enter x");
        scanf("%d", &x);
        if (x <A)
        {
            printf("absent");
        }

        else
        {
            printf("present");
        }

        printf("total sum = %d", sum);
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Lecture OK.
- Boucle `while (n > S || A == 5)`. Condition totalement arbitraire et fausse.
- Condition ne change pas dans la boucle (si ce n'est n qui n'est pas décrémenté, ah si `total sum` ?).
- Boucle infinie.

NOTE FINALE : 05 / 20

Feedback :

- **Appréciation globale : Insuffisant.**
-

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 a, s, x, absent, present;
    int n;
    printf("enter the number of registered students: ");
    scanf("%d", &n);
    printf("enter the minimum attendance required:");
    scanf("%d", &a);
    printf("enter the absence threshold:");
    scanf("%d", &s);
    printf("enter the number of attended session:");
    scanf("%d", &x);
    if (x < a) {
        printf("the student is absent");
        scanf("%d", &absent);
    } else {
        printf("the student is present");
        scanf("%d", &present);
        n++;
    }
    printf("the number of present student is: %d", present);
    printf("the number of absent student is: %d", absent);
    return 0;
}
```

Copy 14

```
#include <stdio.h>
int main()
{
    int a, s, x, absent, present;
    int n;
    printf("enter the number of registered students: ");
    scanf("%d", &n);
    printf("enter the minimum attendance required: ");
    scanf("%d", &a);
    printf("enter the absence threshold: ");
    scanf("%d", &s);
    printf("enter the number of attended session: ");
    scanf("%d", &x);
    if (x < a)
    {
        printf("the student is absent");
        absent = n + 1;
    }

    else
    {
        printf("the student is present");
        present = n + 1;
    }

    printf("the number of present student is: %d", present);
    printf("the number of absent student is: %d", present);
    return 0;
}
```

Analyse :

Algorithmique :

- Lecture de x avant la boucle.
- Pas de boucle !! Code séquentiel.
- `absent = n + 1`. Formule magique ?

NOTE FINALE : 03 / 20

Feedback :

- **Appréciation globale : Très Insuffisant.** Pas d'algorithme.
-

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, X;
    int i = 1, present = 0, absent = 0;
    printf("Please input the total number of registered students: ");
    scanf("%d", &N);
    printf("Please input the minimum attendance required: ");
    scanf("%d", &A);
    printf("Please input the absence threshold: ");
    scanf("%d", &S);
    printf("\n Please input the number of attended sessions for each student:\n\n");
    for (i = 1; i <= N; i++) {
        if (absent < S) {
            printf("Student number: %02d \n", i);
            printf("X: ");
            scanf("%d", &X);
            printf("Student status: ");
            if (X < A) {
                absent++;
                printf("Absent \n");
            }
            else {
                present++;
                printf("Present \n");
            }
            printf("Present students: %d \n", present);
            printf("Absent students: %d \n", absent);
        }
        printf("Total processed students: %d \n", present + absent);
        printf("Present students: %d \n", present);
        printf("Absent students: %d \n", absent);
        printf("Final status: \n");
        if (absent < S) { printf("Session Valid"); }
        else { printf("Session Cancelled"); }
    }
    return 1;
}

```

```
#include <stdio.h>
int main()
{
    int N, A, S, X;
    int i = 1, present = 0, absent = 0;
    printf("Please input the total number of registered students: ");
    scanf("%d", &N);
    printf("Please input the minimum attendance required: ");
    scanf("%d", &A);
    printf("Please input the absence threshold: ");
    scanf("%d", &S);
    printf("Please input the number of attended sessions for each student: ");
    for (i = 1; i <= N; i++)
    {
        if (absent < S)
        {
            printf("Student number: %02d", i);
            printf("X: ");
            scanf("%d", &X);
            printf("Student status: ");
            if (X < A)
            {
                absent++;
                printf("Absent");
            }

            else
            {
                present++;
                printf("Present");
            }

            printf("Present students: %d", present);
            printf("Absent students: %d", absent);
        }

    }

    printf("Total processed students: %d", present + absent);
    printf("Present students: %d", present);
    printf("Absent students: %d", absent);
    if (absent < S)
    {
        printf("Session Valid");
    }

    else
    {
        printf("Session cancelled");
    }
}
```

```
}  
    return 0;  
}
```


Analyse :

Algorithmique :

- Boucle for sur N.
- A l'intérieur, if (absent < S). C'est une façon valide de gérer l'arrêt (ne plus rien faire si seuil atteint).
- Logique correcte.
- Affichage Student number : %02d. Soigné.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Condition boucle	3 / 4	Arrêt géré par un if englobant. Fonctionnel.
Logique prés./abs.	4 / 4	OK.
Compteurs	3 / 3	OK.
Affichages inter.	2 / 2	OK.
Affichage final	1 / 1	OK.

NOTE FINALE : 19 / 20

Feedback :

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

```

#include <stdio.h>
int main ( ) {
    int N, A, S, X, i = 1, Present = 0, absent = 0;
    printf("entre N~");
    scanf("%d", &N);
    printf("entre A~");
    scanf("%d", &A);
    printf("entre S~");
    scanf("%d", &S);
    while (i <= N && absent < S) {
        printf("Student %d - attended session", i);
        scanf("%d", &X);
        if (X < A) {
            absent++;
        } else {
            Present++;
        }
        printf("step %d", i);
        printf("Present = %d", Present);
        printf("absent = %d", absent);
        i++;
    }
    printf("final result");
    printf("Processed student", i-1);
    printf("absent student", absent);
    printf("Present student", Present);
    if (absent >= S) {
        printf("session cancelled");
    } else {
        printf("session valid");
    }
    return 0;
}

```

Copy 16

```
#include <stdio.h>
int main()
{
    int N, A, S, X, i = 1, present = 0, absent = 0;
    printf("entre N");
    scanf("%d", &N);
    printf("entre A");
    scanf("%d", &A);
    printf("entre S");
    scanf("%d", &S);
    while (i <= N && absent < S)
    {
        printf("Student %d - attended session", i);
        scanf("%d", &X);
        if (X < A)
        {
            absent++;
        }

        else
        {
            present++;
        }

        printf("step %d", i);
        printf("present = %d", present);
        printf("absent = %d", absent);
        i++;
    }

    printf("final result");
    printf("processed student %d", i - 1);
    printf("absent student %d", absent);
    printf("present student %d", present);
    if (absent == S)
    {
        printf("session cancelled");
    }

    else
    {
        printf("session Valid");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Boucle while (`i <= N && absent < S`). Correcte.
- Logique parfaite.
- Affichage final correct.

NOTE FINALE : 20 / 20

Feedback :

- **Appréciation globale : Excellent.**
-

```
#include <stdio.h>
```

```
read A, N, S
i = 1
```

```
present = 0
```

```
absent = 0
```

```
while i ≤ N and absent < S =
```

```
read x
```

```
if x < A =
```

```
else =
```

```
present ++
```

```
print i, present, absent
```

```
i ++
```

```
if absent == S :
```

```
Session cancelled
```

```
else :
```

```
Session Vaild
```

```
{ return 0;
```

Copy 17

```
#include <stdio.h>
int main()
{
    // read A, N, S
    int i = 1;
    int present = 0;
    int absent = 0;
    while (i <= N && absent < S)
    {
        // read x if (x < A)
        {
            absent++;
        }

        else
        {
            present++;
        }

        // print i, present, absent
        i++;
    }

    if (absent == S)
    {
        printf("session cancelled");
    }

    else
    {
        printf("session valid");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Code minimaliste (commentaires `// read A` au lieu du code).
- Logique `while` correcte.
- Incomplet sur la lecture.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	0 / 3	Commentaires au lieu de code.
Initialisation	3 / 3	OK.
Condition boucle	4 / 4	OK.
Logique prés./abs.	3 / 4	x non lu explicitement (implicite).
Compteurs	3 / 3	OK.
Affichage final	1 / 1	OK.

NOTE FINALE : 14 / 20

Feedback :

- **Appréciation globale : Moyen.** Code fonctionnel mais les E/S sont simulées par des commentaires.
-


```

#include <stdio.h>

int main () {
    int N, A, X, i;
    constant S;
    For (i = 1, i++) {
        printf("read the number of attended sessions x");
        if (x < A) {
            printf("student absent");
            scanf("%d", student absent);
        } else if (x > A) {
            printf("student present");
            scanf("%d", student present);
            number of present student = N - student absent;
            number of absent student = N - student present;
            scanf("%d %d", number of present student, number of absent student);
            return 1;
            if (A == N) {
                printf("stop");
            } else if (N == S) {
                printf("stop");
            }
            if ("the student is present") {
                printf("session valid");
            } else if ("the student is absent") {
                printf("session cancelled");
            }
        }
        return 0;
    }
}

```


Copy 18

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

        else if (x > A)
        {
            printf("student present");
            scanf("%d", &student_present);
        }

        number_of_present_student = N - student_absent;
        number_of_absent_student = N - student_present;
        scanf("%d %d", &number_of_present_student, &number_of_absent_student);
        if (A == N)
        {
            printf("stop");
        }

        else if (N == S)
        {
            printf("stop");
        }

        if (the student is present)
        {
            printf("session valid");
        }

        else if (the student is absent)
        {
            printf("session cancelled");
        }

    }

    return 0;
}
```

Analyse :

Algorithmique :

- Boucle for.
- Logique interne : `if (x < A)` mais x non lu (pas de `scanf`).
- `if (X < A)` lit `student_absent` via `scanf` ? L'étudiant saisit le nombre d'absents ?? Contresens.
- Condition arrêt `if (A == N)` dans la boucle.

NOTE FINALE : 07 / 20

Feedback :

- **Appréciation globale : Insuffisant.**
-

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, S, A;

    printf("enter num of students");
    scanf("%d", &N);
    printf("min attendance reqd");
    scanf("%d", &A);
    printf("absence threshold");
    scanf("%d", &S);
    int P=0, D=0;
    for (int i=1; i<=N; i++) {
        printf("enter num of attendance for student %d", i);
        scanf("%d", &x);
        printf("Num of students are %d", i);
        if (x < A) { P++;
            printf("students are absents", P); }
        else { D++;
            printf("%d students are present", D); }
    }

    int T=0;
    T = D + P;
    printf("Total students is %d", T);
    printf("Present students are: %d, D");
    printf("absent students are: %d, P");
    if (T >= S) printf("session valid");
    else printf("session cancelled");

    return 0; }

```

```
#include <stdio.h>
int main()
{
    int N, S, A;
    printf("enter num of students");
    scanf("%d", &N);
    printf("min attendance red");
    scanf("%d", &A);
    printf("absence threshold");
    scanf("%d", &S);
    int i, x, P = 0, D = 0;
    for (int i = 1; i <= N; i++)
    {
        printf("enter num of attendance for student %d", i);
        scanf("%d", &x);
        printf("Num of students are %d", i);
        if (x < A)
        {
            P++;
            printf("students are absents");
        }

        else
        {
            D++;
            printf("%d students are present", D);
        }

    }

    int T = 0;
    T = D + P;
    printf("total students is %d", T);
    printf("present students are %d", D);
    printf("absent students are %d", P);
    if (T >= S)
    {
        printf("session valid");
    }

    else
    {
        printf("session cancelled");
    }

    return 0;
}
```

Analyse :**Algorithmique :**

- Lectures correctes.
- Boucle for sur N avec lecture interne correcte.
- Logique interne correcte (compteurs P, D).
- Affichage final calculé.
- Condition finale `if (T >= S)` compare le nombre total d'étudiants au seuil ? Confusion. Devrait être P (absents selon sa logique P++ si `x < A`).

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Initialisation	3 / 3	OK.
Condition boucle	2 / 4	Ok mais pas d'arrêt sur seuil.
Logique prés./abs.	4 / 4	OK.
Compteurs	3 / 3	OK.
Affichages inter.	2 / 2	OK.
Affichage final	0 / 1	Logique finale fausse.

NOTE FINALE : 17 / 20**Feedback :**

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

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 v, A, S;
    int N (total number of reg student);
    printf("enter total number registered student N and student number");
    scanf("%d", &N);
    write("The student is absent");
    if (number of attended session < A;
    {
        write("The student is present");
    }
    if (number of attended session > A;
    {
        printf("enter the number of attended sessions x student by student");
        scanf("%d", &x);
        if (x == N) // total present student
        {
            write("simulation stop");
        }
    }
}
```

```
printf("enter the number of attended sessions x student by student");
scanf("%d", &x);
write("session valid");
if (x == N)
{
    write("session canceled");
}
if (x > N)
{
    scanf("%d", &x);
}
return 0;
}
```

```
#include <stdio.h>
int main()
{
    int N, A, S;
    int total_number_of_registered;
    printf("enter total number of registered students");
    scanf("%d", &N);
    if (number_of_attended_sessions < A)
    {
        write("the student is absent");
        write("number of attended session");
    }

    else if (number_of_attended_sessions > A)
    {
        write("the student is present");
    }

    printf("enter the number of attended sessions x student by student");
    scanf("%d %d", &present_students, &absent_students);
    write("simulations stop");
    if (x == N)
    {
        // total processed student if (absent < S)
        {
            write("session valid");
        }

    }

    if (absent > S)
    {
        write("session cancelled");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Utilise `write` au lieu de `printf`. Pseudo-code.
- Variables très longues explicites (`total_number . . .`) mais utilisées sans déclaration ou lecture adéquate dans une boucle.
- Pas de boucle explicite autour de la logique principale (seulement des `if` séquentiels).
- `if (x == N)` à la fin ?

NOTE FINALE : 05 / 20

Feedback :

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

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, X, aps, ress;
    printf("entre a total number of registered student");
    scanf("%d", &N);
    printf("entre minimum attendance required");
    scanf("%d", &A);
    printf("entre absence the reshould");
    scanf("%d", &S);
    printf("entre the mber of attended session");
    scanf("%d", &X);
    printf("entre aps");
    scanf("%d", &aps);
    printf("entre ress");
    scanf("%d", &ress);
    if (N < A){
        printf("the student is consideral absent");
    }
    else {
        printf("the student is present");
        while (A == N && A == S) {
            if (X < A) {
                S = N - 1;
            }
            else {
                X++;
            }
        }
        if (aps < S && ress >= A) {
            printf("session valide");
        }
        else {
            printf("session non valide");
        }
    }
}

```

```
#include <stdio.h>
int main()
{
    int N, A, S, X, aps, ress;
    printf("entre a total number of registered student");
    scanf("%d", &N);
    printf("entre minimum attendance required");
    scanf("%d", &A);
    printf("entre absence threshold");
    scanf("%d", &S);
    printf("entre number of attended session");
    scanf("%d", &X);
    printf("entre aps");
    scanf("%d", &aps);
    printf("entre ress");
    scanf("%d", &ress);
    if (n < A)
    {
        printf("the student is considered absent");
    }

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

    while (A == N && A == S)
    {
        if (X < A)
        {
            S = N - 1;
        }

        else
        {
            X++;
        }
    }

    if (aps < S && ress >= A)
    {
        printf("session valide");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Lectures.
- Logic if (n < A) hors boucle.
- Boucle while (A == N && A == S) ? Condition quasi-impossible.
- Contenu incohérent.

NOTE FINALE : 03 / 20

Feedback :

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

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;
    int A;
    int S;

    printf("Enter Total number of registered students : ");
    scanf("%d", &N);

    printf("Enter number of minimum attendance required : ");
    scanf("%d", &A);

    printf("Enter number of absence threshold : ");
    scanf("%d", &S);

    for(int i = 1; i <= N; i++) while(i <= N || B == S) {
        int i = 1;
        i++;
        int x;
        int B = 0;
        int P = 0;

        printf("Enter the number of absence of student no %d: ", i);
        scanf("%d", &x);
        if(i == N || B == S) {
            printf("Total processed students is: %d\n", i);
            printf("Present students is: %d\n", P);
            printf("Absent students is: %d\n", B);
            if(B == S) {
                printf("Session status is canceled\n");
            } else { printf("Session status is valid\n"); }
            if(x < A) {
                B++;
            } else { P++; }
        }
        return 0;
    }

```

```
#include <stdio.h>
int main()
{
    int N, A, S;
    printf("Enter Total number of registered students: ");
    scanf("%d", &N);
    printf("Enter number of minimum attendance required: ");
    scanf("%d", &A);
    printf("Enter number of absence threshold: ");
    scanf("%d", &S);
    while (i <= N || B == S)
    {
        int B = 0;
        int P = 0;
        printf("Enter the number of absence of student n %d", i);
        scanf("%d", &X);
        if (i == N || B == S)
        {
            printf("Total processed students is %d", i);
            printf("Present students is %d", P);
            printf("Absent students is %d", B);
            if (B == S)
            {
                printf("Session status is cancelled");
            }

            else
            {
                printf("Session status is valid");
            }
        }

        if (X < A)
        {
            B++;
        }

        else
        {
            P++;
        }
    }

    return 0;
}
```

Analyse :**Algorithmique :**

- Initialisation `int i` manquante avant `while`.
- Boucle `while (i <= N || B == S)`. Condition arrêt fausse (`||` continue si seuil atteint).
- Réinitialisation `int B=0; int P=0; DANS la boucle !` Les compteurs sont remis à zéro à chaque tour.
- Logique d'incrémentation en fin de boucle. `i` n'est jamais incrémenté (boucle infinie).

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Initialisation	0 / 3	Compteurs remis à 0 dans la boucle. <code>i</code> non init.
Condition boucle	0 / 4	Boucle infinie (<code>i</code> ne change pas).
Logique prés./abs.	4 / 4	OK.
Compteurs	0 / 3	Remis à zéro.
Affichage final	0 / 1	-

NOTE FINALE : 07 / 20**Feedback :**

- **Appréciation globale : Insuffisant.** Erreur sur la portée des variables.