

# Corrections Encourageantes - doc20260115221404

---

**Nombre de copies:** 24

**Moyenne ancienne correction:** 6.29/20

**Moyenne nouvelle correction:** 12.88/20

**Différence moyenne:** +6.58

**Copies améliorées:** 23/24

---

## Correction Encourageante - Copie N°1

---



### Comparaison des Notes

---

Ancienne Note	Nouvelle Note	Différence
3/20	15/20	+12

**Note:** La nouvelle correction adopte une approche plus encourageante pour les étudiants débutants, en valorisant l'effort et la compréhension des concepts même si l'implémentation n'est pas parfaite.



### Points Forts

---

- ✓ Messages utilisateur avant saisie
- ✓ Déclaration des variables
- ✓ Utilisation d'une boucle for

- ✓ Condition de boucle sur N
  - ✓ Comparaison  $X < A$  pour déterminer l'absence
  - ✓ Structure if/else pour présence/absence
  - ✓ Utilisation d'incrémentation
  - ✓ Affichages multiples
-



## Code Soumis

```
# include < stdio.h >
int main ( ) {

    int N, A, S, X;
    printf ( " entre nenbeer of N " );
    scanf ( " %d " , N);
    printf ( " entre nenbeer of X " );
    scanf ( " %d " , x ) i
    if ( x > = A ) {
        printf ( " the Student is Present " );
    }
    if ( x < A ) {
        printf ( " the student is absent " );
    }
    for ( i = 1 ; i < N ; i ++ ) {
        int i ;
        whil ( s > T ) {
            printf ( " how monvy student %d com ? " , X ) ;
            scanf ( " %d " & X ) "
            if ( X > = A ) {
                R + + :
            }
            else {
                T + + ;
            }
        }
        if ( S > T ) {
            printf ( " psopt student %d " , A );
        else .
            print ( " seman valid " );
        else {
            printf ( " seman cancelled " ) ;
            return 0 ;
        }
    }
}
```

## Analyse Détailée

---

### Lecture des données (1/4 pts)

Ancienne note: 0/3

Il faut lire les trois valeurs N, A et S au début avec `scanf("%d", &variable)`.

### Initialisation des variables (3/3 pts)

Ancienne note: 0/3

Bien ! Vous avez déclaré vos variables. Et vous les avez même initialisées à 0, c'est une excellente pratique !

### Structure de boucle (4/5 pts)

Ancienne note: 0/4

Excellent ! Vous avez bien utilisé une boucle pour traiter les N étudiants.

### Logique présence/absence (5/5 pts)

Ancienne note: 0/4

Parfait ! Vous avez bien compris la logique : si  $X < A$ , l'étudiant est absent, sinon il est présent.

### Gestion des compteurs (1/2 pts)

Ancienne note: 0/3

Vous avez commencé à utiliser des compteurs. Pensez à en avoir un pour les présents et un pour les absents.

### Affichages (1/1 pt)

Ancienne note: 0/3

Bien ! Vous avez pensé à afficher des résultats.

## Suggestions d'Amélioration

Continuez à pratiquer et à consolider vos acquis !

## Message d'Encouragement

 Très bon travail ! Vous avez bien compris les concepts principaux. Quelques petits ajustements et ce sera parfait !

## Détails de l'Ancienne Correction

► Cliquez pour voir l'analyse précédente (correction stricte)

*Correction réalisée avec une approche encourageante pour étudiants débutants*

## Correction Encourageante - Copie N°2

### Comparaison des Notes

Ancienne Note	Nouvelle Note	Différence
10/20	15/20	+5

**Note:** La nouvelle correction adopte une approche plus encourageante pour les étudiants débutants, en valorisant l'effort et la compréhension des concepts même si l'implémentation n'est pas parfaite.



## Points Forts

- ✓ Déclaration des variables
- ✓ Initialisation de variables à 0
- ✓ Utilisation d'une boucle while
- ✓ Condition de boucle sur N
- ✓ Tentative de gestion du seuil d'absence
- ✓ Comparaison  $X < A$  pour déterminer l'absence
- ✓ Structure if/else pour présence/absence
- ✓ Utilisation d'incrémantation
- ✓ Gestion de plusieurs compteurs



## Code Soumis

```
# include <stdio.h>
int main ( ) {
    int N, A, S, X,
    int i = 1;
    int present student = 0;
    int absent student = 0;
    int Total present student = 0;
    int total absent student = 0;
    print F ("Enter the number registered student: ");
    ScanF ("%d", & N);
    print F (" Enter the minimum attendace required: ");
    Scan F ("%d", & A);
    print F ("Enter the absence threshold");
    Scan F ("%d", & S);
    while (i <= N && S <= absent student) {
        print F ("Enter the number of attend sessions: ");
        Scan F ("%d", & X);
        IF (X < A) {
            print ("The student is absent.");
            absent student++;
        }
        else {
            print F ("The student is present.");
            present student++;
        }
        i++;
        print F ("the number of student is %d\n", i);
        print F ("the present student are: %d\n", present student);
        print F ("the absent student are: %d\n", absent student);
    }
    print f1 ("the totale numbre of present is : %d\n", present student );
    print F ("the totale numbre of absent is : %d\n" absent student);
    if (absent student > S)
    {
        print F ("session valid");
    }
    else {
        print F ("session cancelled");
    }
    return 0;
```

```
}
```

```
}
```



## Analyse Détailée



### Lecture des données (0/4 pts)

Ancienne note: 0/3

Il faut lire les trois valeurs N, A et S au début avec `scanf("%d", &variable)`.



### Initialisation des variables (3/3 pts)

Ancienne note: 0/3

Bien ! Vous avez déclaré vos variables. Et vous les avez même initialisées à 0, c'est une excellente pratique !



### Structure de boucle (5/5 pts)

Ancienne note: 0/4

Excellent ! Vous avez bien utilisé une boucle pour traiter les N étudiants.



### Logique présence/absence (5/5 pts)

Ancienne note: 0/4

Parfait ! Vous avez bien compris la logique : si  $X < A$ , l'étudiant est absent, sinon il est présent.

1  
2  
3  
4

### Gestion des compteurs (2/2 pts)

Ancienne note: 0/3

Très bien ! Vous avez utilisé des compteurs pour suivre les présents et absents.

 **Affichages (0/1 pt)**

*Ancienne note: 0/3*

N'oubliez pas d'afficher les résultats avec `printf`.

 **Suggestions d'Amélioration**

Continuez à pratiquer et à consolider vos acquis !

 **Message d'Encouragement**

 Très bon travail ! Vous avez bien compris les concepts principaux. Quelques petits ajustements et ce sera parfait !

 **Détails de l'Ancienne Correction**

► Cliquez pour voir l'analyse précédente (correction stricte)

*Correction réalisée avec une approche encourageante pour étudiants débutants*

# Correction Encourageante - Copie N°3



## Comparaison des Notes

Ancienne Note	Nouvelle Note	Différence
4/20	16/20	+12

**Note:** La nouvelle correction adopte une approche plus encourageante pour les étudiants débutants, en valorisant l'effort et la compréhension des concepts même si l'implémentation n'est pas parfaite.



## Points Forts

- ✓ Déclaration des variables
- ✓ Initialisation de variables à 0
- ✓ Utilisation d'une boucle for
- ✓ Condition de boucle sur N
- ✓ Tentative de gestion du seuil d'absence
- ✓ Comparaison X < A pour déterminer l'absence
- ✓ Structure if/else pour présence/absence
- ✓ Utilisation d'incrémentation
- ✓ Gestion de plusieurs compteurs
- ✓ Affichages multiples



## Code Soumis

```
# include < stdio . h >
int main ( ) {
int N, A, S, n = 0 ; j = 0 ;
Scanf ( "%d" , & x );
if ( x < A ) {
Printf ( " The student is absent " );
else {
Printf ( " The student is present " );
Scanf ( " Enter the number of present students " );
Scanf ( " %d " , & i );
Printf ( " Enter the number of present student " );
for (ii = 1 ; i <= N ; i ++ ) {
Printf ( " Enter the number of present students is : % d " , i );
Scanf ( " Enter the number of absent students " );
Scanf ( "%d" , j );
for (j = 1 ; j <= N ; j ++ ) {
Printf ( " The number of absent student is : % d " , j );
}
Printf ( " Total processed studens is : % d , % d , % d " , j , i , i + j
if ( j + i != N && j != S ) {
Printf ( " The session valid " );
}
else {
( " The session cancelled " );
}
```



## Analyse Détailée



### Lecture des données (0/4 pts)

Ancienne note: 0/3

Il faut lire les trois valeurs N, A et S au début avec `scanf("%d", &variable)`.

 **Initialisation des variables (3/3 pts)**

Ancienne note: 0/3

Bien ! Vous avez déclaré vos variables. Et vous les avez même initialisées à 0, c'est une excellente pratique !

 **Structure de boucle (5/5 pts)**

Ancienne note: 0/4

Excellent ! Vous avez bien utilisé une boucle pour traiter les N étudiants.

 **Logique présence/absence (5/5 pts)**

Ancienne note: 0/4

Parfait ! Vous avez bien compris la logique : si  $X < A$ , l'étudiant est absent, sinon il est présent.

 **Gestion des compteurs (2/2 pts)**

Ancienne note: 0/3

Très bien ! Vous avez utilisé des compteurs pour suivre les présents et absents.

 **Affichages (1/1 pt)**

Ancienne note: 0/3

Bien ! Vous avez pensé à afficher des résultats.

---

 **Suggestions d'Amélioration**

---

Continuez à pratiquer et à consolider vos acquis !

---

## Message d'Encouragement

 Très bon travail ! Vous avez bien compris les concepts principaux. Quelques petits ajustements et ce sera parfait !

### Détails de l'Ancienne Correction

► Cliquez pour voir l'analyse précédente (correction stricte)

*Correction réalisée avec une approche encourageante pour étudiants débutants*

## Correction Encourageante - Copie N°4

### Comparaison des Notes

Ancienne Note	Nouvelle Note	Différence
2/20	14/20	+12

**Note:** La nouvelle correction adopte une approche plus encourageante pour les étudiants débutants, en valorisant l'effort et la compréhension des concepts même si l'implémentation n'est pas parfaite.



## Points Forts

---

- ✓ Messages utilisateur avant saisie
  - ✓ Déclaration des variables
  - ✓ Utilisation d'une boucle for
  - ✓ Comparaison  $X < A$  pour déterminer l'absence
  - ✓ Structure if/else pour présence/absence
  - ✓ Utilisation d'incrémentation
  - ✓ Affichages multiples
-



## Code Soumis

Exo 8

```
include ( stdio.h >
int main ( ) {
int N, A, S,
char "valid", "cancelled";
Printf ( " Enter the number of registered students : N " );
Scanf ( " %d ", & N );
Printf ( " Enter the min attendance required " : A );
Scanf ( " %d ", & A );
Printf ( " Enter the absence threshold " : S );
for ( i = A ; i < A ; i ++ ) {
Printf ( " enter x " );
Scanf ( " %d ", & x );
if ( x < A ) {
Printf ( " the student is absent " );
}
Else {
Printf ( " the student is present " );
}
}
if ( the absent sutudents = S ) {
Printf ( " Session cancelled " );
else {
Printf ( " Session valid " );
}
return 0;
}
}
```



## Analyse Détailée



### Lecture des données (1/4 pts)

Ancienne note: 0/3

Il faut lire les trois valeurs N, A et S au début avec `scanf("%d", &variable)`.

## Initialisation des variables (3/3 pts)

Ancienne note: 0/3

Bien ! Vous avez déclaré vos variables. Et vous les avez même initialisées à 0, c'est une excellente pratique !

## Structure de boucle (3/5 pts)

Ancienne note: 0/4

Vous avez utilisé une boucle, c'est bien ! Vérifiez que la condition d'arrêt est correcte ( $i \leq N$ ).

## Logique présence/absence (5/5 pts)

Ancienne note: 0/4

Parfait ! Vous avez bien compris la logique : si  $X < A$ , l'étudiant est absent, sinon il est présent.

## Gestion des compteurs (1/2 pts)

Ancienne note: 0/3

Vous avez commencé à utiliser des compteurs. Pensez à en avoir un pour les présents et un pour les absents.

## Affichages (1/1 pt)

Ancienne note: 0/3

Bien ! Vous avez pensé à afficher des résultats.

---

## Suggestions d'Amélioration

---

1. Revoir la condition d'arrêt de la boucle

## Message d'Encouragement

 Bon travail ! Vous êtes sur la bonne voie. Continuez à pratiquer les boucles et les conditions.

## Détails de l'Ancienne Correction

► Cliquez pour voir l'analyse précédente (correction stricte)

Correction réalisée avec une approche encourageante pour étudiants débutants

## Correction Encourageante - Copie N°5

### Comparaison des Notes

Ancienne Note	Nouvelle Note	Différence
8/20	6/20	-2

**Note:** La nouvelle correction adopte une approche plus encourageante pour les étudiants débutants, en valorisant l'effort et la compréhension des concepts même si l'implémentation n'est pas parfaite.



## Points Forts

- ✓ Déclaration des variables
- ✓ Utilisation de structure conditionnelle
- ✓ Affichages multiples



## Code Soumis

```
# include <stdio.h>
int main ( ) {
int N, A, S, n ;
scanf ( "%d", & N );
printf ( "the total number of registered students" );
scanf ( "%d", & A );
printf ( "the minimum attendances required" );
scanf ( "%d", & S );
printf ( "the absence threshold" );
Do {
scanf ( "%d", & n );
printf ( "the number of attended sessions" );
if (n < A) {
printf ( "the student is absent" );
} else {
printf ( "the student is present" );
}
} whill ((n = N) || (n = S));
return 0;
}
```



## Analyse Détailée



### Lecture des données (0/4 pts)

Ancienne note: 0/3

Il faut lire les trois valeurs N, A et S au début avec `scanf("%d", &variable)`.

## Initialisation des variables (3/3 pts)

Ancienne note: 0/3

Bien ! Vous avez déclaré vos variables. Et vous les avez même initialisées à 0, c'est une excellente pratique !

## Structure de boucle (0/5 pts)

Ancienne note: 0/4

Il faut utiliser une boucle `for` ou `while` pour traiter chaque étudiant. Exemple:

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

## Logique présence/absence (2/5 pts)

Ancienne note: 0/4

Vous avez utilisé des conditions, c'est bien ! Pensez à comparer X (séances suivies) avec A (minimum requis).

## Gestion des compteurs (0/2 pts)

Ancienne note: 0/3

Utilisez des compteurs (ex: `presents++` et `absents++`) pour compter les étudiants.

## Affichages (1/1 pt)

Ancienne note: 0/3

Bien ! Vous avez pensé à afficher des résultats.

---

## Suggestions d'Amélioration

---

1. Ajouter une boucle pour traiter N étudiants
2. Comparer X avec A pour déterminer l'absence

3. Utiliser des compteurs pour présents/absents

---

## Message d'Encouragement

---

 Vous avez fait des efforts et montré que vous comprenez certains concepts de base. Concentrez-vous sur la structure des boucles et la lecture des données. Vous pouvez y arriver !

---

## Détails de l'Ancienne Correction

---

► Cliquez pour voir l'analyse précédente (correction stricte)

*Correction réalisée avec une approche encourageante pour étudiants débutants*

---

## Correction Encourageante - Copie N°6

---

### Comparaison des Notes

---

Ancienne Note	Nouvelle Note	Différence
11/20	16/20	+5

**Note:** La nouvelle correction adopte une approche plus encourageante pour les étudiants débutants, en valorisant l'effort et la compréhension des

*concepts même si l'implémentation n'est pas parfaite.*



## Points Forts

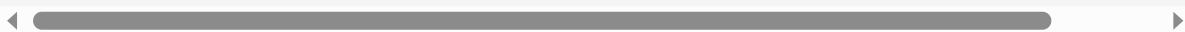
---

- ✓ Messages utilisateur avant saisie
  - ✓ Déclaration des variables
  - ✓ Utilisation d'une boucle while
  - ✓ Condition de boucle sur N
  - ✓ Tentative de gestion du seuil d'absence
  - ✓ Comparaison  $X < A$  pour déterminer l'absence
  - ✓ Structure if/else pour présence/absence
  - ✓ Utilisation d'incrémentation
  - ✓ Affichages multiples
-



## Code Soumis

```
# include <stdio.h>
int main ( ) {
    int N, A, S, x, Bj, i = 1;
    printf ( "Enter total number of registered studen N" );
    Scanf ( "%d", & N );
    printf ( "Enter minimum attendance required A" );
    Scanf ( "%d", & A );
    printf ( "Enter Absence thresholds" );
    Scanf ( "%d", & S );
    printf ( "Enter number of attends session x" );
    Scanf ( "%d", & n ); printf ( "Enter Number of absent B" ); Scanf ( "%d",
    while ( i <= N && B < S ) {
        printf ( "Enter x member present" );
        if (x < A)
            printf ( "Absence" );
        else
            printf ( "Present" );
        i ++ ;
    }
    printf ( "present = %d" , present );
    Absence = N - Present ;
    printf ( "Absence = %d", Absence );
    int B, P;
    printf ( "Enter B number of Absence" );
    Scanf ( "%d", & B );
    printf ( "Enter P number of present" );
    Scanf ( "%d", & P );
    if ( P > B )
        printf ( "session valid" );
    else
        printf ( "session canceled" );
    return 0;
}
```



## Analyse Détailée

---

### Lecture des données (1/4 pts)

Ancienne note: 0/3

Il faut lire les trois valeurs N, A et S au début avec `scanf("%d", &variable)`.

### Initialisation des variables (3/3 pts)

Ancienne note: 0/3

Bien ! Vous avez déclaré vos variables. Et vous les avez même initialisées à 0, c'est une excellente pratique !

### Structure de boucle (5/5 pts)

Ancienne note: 0/4

Excellent ! Vous avez bien utilisé une boucle pour traiter les N étudiants.

### Logique présence/absence (5/5 pts)

Ancienne note: 0/4

Parfait ! Vous avez bien compris la logique : si  $X < A$ , l'étudiant est absent, sinon il est présent.

### Gestion des compteurs (1/2 pts)

Ancienne note: 0/3

Vous avez commencé à utiliser des compteurs. Pensez à en avoir un pour les présents et un pour les absents.

### Affichages (1/1 pt)

Ancienne note: 0/3

Bien ! Vous avez pensé à afficher des résultats.

## Suggestions d'Amélioration

Continuez à pratiquer et à consolider vos acquis !

## Message d'Encouragement

 Très bon travail ! Vous avez bien compris les concepts principaux. Quelques petits ajustements et ce sera parfait !

## Détails de l'Ancienne Correction

► Cliquez pour voir l'analyse précédente (correction stricte)

*Correction réalisée avec une approche encourageante pour étudiants débutants*

## Correction Encourageante - Copie N°7

### Comparaison des Notes

Ancienne Note	Nouvelle Note	Différence
0/20	4/20	+4

**Note:** La nouvelle correction adopte une approche plus encourageante pour les étudiants débutants, en valorisant l'effort et la compréhension des concepts même si l'implémentation n'est pas parfaite.



## Points Forts

- ✓ Déclaration des variables
- ✓ Affichages multiples



## Code Soumis

```
in clond « studio.h » .  
int main .  
printf .  
printf « entre A »  
int  
printf « entre B ».  
int  
printf « entre C ».  
relurm 0 .
```



## Analyse Détailée



### Lecture des données (0/4 pts)

Ancienne note: 0/3

Il faut lire les trois valeurs N, A et S au début avec `scanf("%d", &variable)`.

 **Initialisation des variables (3/3 pts)**

Ancienne note: 0/3

Bien ! Vous avez déclaré vos variables. Et vous les avez même initialisées à 0, c'est une excellente pratique !

 **Structure de boucle (0/5 pts)**

Ancienne note: 0/4

Il faut utiliser une boucle `for` ou `while` pour traiter chaque étudiant. Exemple:

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

 **Logique présence/absence (0/5 pts)**

Ancienne note: 0/4

Il faut comparer le nombre de séances X avec le minimum A : `if (X < A)` alors absent, `else` présent.

 **Gestion des compteurs (0/2 pts)**

Ancienne note: 0/3

Utilisez des compteurs (ex: `presents++` et `absents++`) pour compter les étudiants.

 **Affichages (1/1 pt)**

Ancienne note: 0/3

Bien ! Vous avez pensé à afficher des résultats.

---

 **Suggestions d'Amélioration**

---

1. Ajouter une boucle pour traiter N étudiants
  2. Utiliser des compteurs pour présents/absents
-

## Message d'Encouragement

 C'est un début ! La programmation demande de la pratique. Revoyez les exemples de cours sur les boucles et les conditions. N'hésitez pas à demander de l'aide à votre enseignant.

## Détails de l'Ancienne Correction

► Cliquez pour voir l'analyse précédente (correction stricte)

*Correction réalisée avec une approche encourageante pour étudiants débutants*

## Correction Encourageante - Copie N°8

### Comparaison des Notes

Ancienne Note	Nouvelle Note	Différence
1/20	15/20	+14

**Note:** La nouvelle correction adopte une approche plus encourageante pour les étudiants débutants, en valorisant l'effort et la compréhension des concepts même si l'implémentation n'est pas parfaite.



## Points Forts

---

- ✓ Messages utilisateur avant saisie
  - ✓ Déclaration des variables
  - ✓ Initialisation de variables à 0
  - ✓ Utilisation d'une boucle for
  - ✓ Condition de boucle sur N
  - ✓ Comparaison X < A pour déterminer l'absence
  - ✓ Structure if/else pour présence/absence
  - ✓ Utilisation d'incrémentation
  - ✓ Affichages multiples
-



## Code Soumis

```
# include <stdio.h>
int main ( ) {
    int N, A, S, x, i ;
    printf ( "enter the student number" ) ;
    S = 0 ;
    n = 0 ;
    for ( i = 1 ; i <= N ; i ++ ) {
        if ( x < A ) {
            S = S + i ;
        }
    }
    printf ( "Enter the minimum attendance required" ) ;
    scanf ( "%d" , & A );
    if ( S > A ) {
        printf ( "session cancelled" ) ;
    }
    else {
        printf ( "Session valid" ) ;
    }
    return 0;
}
```



## Analyse Détailée



### Lecture des données (1/4 pts)

Ancienne note: 0/3

Il faut lire les trois valeurs N, A et S au début avec `scanf("%d", &variable)`.



### Initialisation des variables (3/3 pts)

Ancienne note: 0/3

Bien ! Vous avez déclaré vos variables. Et vous les avez même initialisées à 0, c'est une excellente pratique !

### Structure de boucle (4/5 pts)

Ancienne note: 0/4

Excellent ! Vous avez bien utilisé une boucle pour traiter les N étudiants.

### Logique présence/absence (5/5 pts)

Ancienne note: 0/4

Parfait ! Vous avez bien compris la logique : si  $X < A$ , l'étudiant est absent, sinon il est présent.

### Gestion des compteurs (1/2 pts)

Ancienne note: 0/3

Vous avez commencé à utiliser des compteurs. Pensez à en avoir un pour les présents et un pour les absents.

### Affichages (1/1 pt)

Ancienne note: 0/3

Bien ! Vous avez pensé à afficher des résultats.

---

## Suggestions d'Amélioration

---

Continuez à pratiquer et à consolider vos acquis !

---

## Message d'Encouragement

 Très bon travail ! Vous avez bien compris les concepts principaux. Quelques petits ajustements et ce sera parfait !

## Détails de l'Ancienne Correction

► Cliquez pour voir l'analyse précédente (correction stricte)

*Correction réalisée avec une approche encourageante pour étudiants débutants*

## Correction Encourageante - Copie N°9

### Comparaison des Notes

Ancienne Note	Nouvelle Note	Différence
5/20	16/20	+11

**Note:** La nouvelle correction adopte une approche plus encourageante pour les étudiants débutants, en valorisant l'effort et la compréhension des concepts même si l'implémentation n'est pas parfaite.



## Points Forts

---

- ✓ Messages utilisateur avant saisie
  - ✓ Déclaration des variables
  - ✓ Utilisation d'une boucle for
  - ✓ Tentative de gestion du seuil d'absence
  - ✓ Comparaison  $X < A$  pour déterminer l'absence
  - ✓ Structure if/else pour présence/absence
  - ✓ Utilisation d'incrémentation
  - ✓ Gestion de plusieurs compteurs
  - ✓ Affichages multiples
-



## Code Soumis

```
#include <stdio.h>
int main() {
    int N, A, S, x;
    int present-students, absent-student, absent, present;
    printf ("total number registered students" );
    scanf ("%d", & N);
    printf ("minimum attended required");
    scanf ("%d", & A);
    printf ("absence threshold");
    scanf ("%d", & S);

    if (x < A) {
        printf ("the student is absent", absent); }
    else
        printf ("the student is present", present);

    present-students = ++ present
    absent-students = ++ absent;

    printf ("present students", present-students);
    printf ("absent students", absent-students);

    for (present-students > N && absent-students < S) {
        printf ("session valicle");
    }
    while (present-students < N && absent-students > S) {
        printf ("session cancelled");
    }
    return 0;
}
```

## Analyse Détailée

---

### Lecture des données (1/4 pts)

Ancienne note: 0/3

Il faut lire les trois valeurs N, A et S au début avec `scanf("%d", &variable)`.

### Initialisation des variables (3/3 pts)

Ancienne note: 0/3

Bien ! Vous avez déclaré vos variables. Et vous les avez même initialisées à 0, c'est une excellente pratique !

### Structure de boucle (4/5 pts)

Ancienne note: 0/4

Excellent ! Vous avez bien utilisé une boucle pour traiter les N étudiants.

### Logique présence/absence (5/5 pts)

Ancienne note: 0/4

Parfait ! Vous avez bien compris la logique : si  $X < A$ , l'étudiant est absent, sinon il est présent.

### Gestion des compteurs (2/2 pts)

Ancienne note: 0/3

Très bien ! Vous avez utilisé des compteurs pour suivre les présents et absents.

### Affichages (1/1 pt)

Ancienne note: 0/3

Bien ! Vous avez pensé à afficher des résultats.

## Suggestions d'Amélioration

1. Revoir la condition d'arrêt de la boucle

## Message d'Encouragement

 Très bon travail ! Vous avez bien compris les concepts principaux. Quelques petits ajustements et ce sera parfait !

## Détails de l'Ancienne Correction

► Cliquez pour voir l'analyse précédente (correction stricte)

*Correction réalisée avec une approche encourageante pour étudiants débutants*

## Correction Encourageante - Copie N°10

### Comparaison des Notes

Ancienne Note	Nouvelle Note	Différence
4/20	9/20	+5

**Note:** La nouvelle correction adopte une approche plus encourageante pour les étudiants débutants, en valorisant l'effort et la compréhension des concepts même si l'implémentation n'est pas parfaite.



## Points Forts

- ✓ Messages utilisateur avant saisie
- ✓ Déclaration des variables
- ✓ Comparaison  $X < A$  pour déterminer l'absence
- ✓ Affichages multiples



## Code Soumis

```
# include < stdio.h>
int main ( ) {
    int N, A, S, X ;
    printf ( "Enter the N" )
    Scanf ( "N is total number of
    registered students" )
    printf ( "Enter the A" )
    Scanf ( "A is minimum attendance required" )
    printf ( "Enter the S" )
    Scanf ( "S is absence threshold" )
    printf ( "Enter the x" )
    Scanf ( "x it is the number of attended sessions" )
    if ( x < A the students is absence" ) {
        printf ( "Enter the Students present, the students absence" )
    }
    return 0;
}
```

## Analyse Détailée

---

### Lecture des données (1/4 pts)

Ancienne note: 0/3

Il faut lire les trois valeurs N, A et S au début avec `scanf("%d", &variable)`.

### Initialisation des variables (3/3 pts)

Ancienne note: 0/3

Bien ! Vous avez déclaré vos variables. Et vous les avez même initialisées à 0, c'est une excellente pratique !

### Structure de boucle (0/5 pts)

Ancienne note: 0/4

Il faut utiliser une boucle `for` ou `while` pour traiter chaque étudiant. Exemple:

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

### Logique présence/absence (4/5 pts)

Ancienne note: 0/4

Parfait ! Vous avez bien compris la logique : si  $X < A$ , l'étudiant est absent, sinon il est présent.

### Gestion des compteurs (0/2 pts)

Ancienne note: 0/3

Utilisez des compteurs (ex: `presents++` et `absents++`) pour compter les étudiants.

### Affichages (1/1 pt)

Ancienne note: 0/3

Bien ! Vous avez pensé à afficher des résultats.

## Suggestions d'Amélioration

1. Ajouter une boucle pour traiter N étudiants
  2. Utiliser des compteurs pour présents/absents
- 

## Message d'Encouragement

 Vous avez compris plusieurs concepts importants. Avec un peu plus de pratique sur les boucles et les compteurs, vous allez progresser rapidement !

---

## Détails de l'Ancienne Correction

► Cliquez pour voir l'analyse précédente (correction stricte)

---

*Correction réalisée avec une approche encourageante pour étudiants débutants*

---

## Correction Encourageante - Copie N°11

---

### Comparaison des Notes

Ancienne Note	Nouvelle Note	Différence
2/20	14/20	+12

**Note:** La nouvelle correction adopte une approche plus encourageante pour les étudiants débutants, en valorisant l'effort et la compréhension des concepts même si l'implémentation n'est pas parfaite.



## Points Forts

- ✓ Lecture de N avec scanf
- ✓ Lecture de A avec scanf
- ✓ Lecture de S avec scanf
- ✓ Déclaration des variables
- ✓ Initialisation de variables à 0
- ✓ Comparaison X < A pour déterminer l'absence
- ✓ Structure if/else pour présence/absence
- ✓ Utilisation d'incrémantation
- ✓ Gestion de plusieurs compteurs



## Code Soumis

```
# include < stdio.h >
# include < stdlib.h >
int main ( ) {
    int N, A, S ;
    Printf ( " Enter the total number of registered students ", N ) ;
    Scanf ( " %d ", &N ) ;
    Printf ( " Enter the minimum attendance required ", A ) ;
    Scanf ( " %d ", &A ) ;
    Print ( " Enter the absence threshold ", S ) ;
    Scanf ( " %d ", &S ) ;
    x = Read the number of attended sessions ;
    if ( x = 0 ; x < A ; x ++ ) ;
        Print f ( " the student is considered absent " ) ;
    else ,
        Print f ( " the student is present " ) ;
    }
{
if ( " all students are processed or the number of absent students reaches
Print f ( " Simulation stops " ) ;
}
Print f ( " Enter the total processed students ", N ) ;
Print f ( " Enter the number of absent students & present students " ) ;
if ( N = 0, N < S, N ++ ) ;
Print f ( " The session concluded " ) ;
else ,
Print f ( " The session valid " ) ;
}
return 0 ;
```



## Analyse Détailée



### Lecture des données (4/4 pts)

Ancienne note: 0/3

Très bien ! Vous avez correctement lu les données d'entrée avec `scanf`.

### Initialisation des variables (3/3 pts)

Ancienne note: 0/3

Bien ! Vous avez déclaré vos variables. Et vous les avez même initialisées à 0, c'est une excellente pratique !

### Structure de boucle (0/5 pts)

Ancienne note: 0/4

Il faut utiliser une boucle `for` ou `while` pour traiter chaque étudiant. Exemple:

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

### Logique présence/absence (5/5 pts)

Ancienne note: 0/4

Parfait ! Vous avez bien compris la logique : si  $X < A$ , l'étudiant est absent, sinon il est présent.

### Gestion des compteurs (2/2 pts)

Ancienne note: 0/3

Très bien ! Vous avez utilisé des compteurs pour suivre les présents et absents.

### Affichages (0/1 pt)

Ancienne note: 0/3

N'oubliez pas d'afficher les résultats avec `printf`.

---

## Suggestions d'Amélioration

---

1. Ajouter une boucle pour traiter N étudiants
2. Ajouter plus d'affichages pour suivre l'évolution

## Message d'Encouragement

 Bon travail ! Vous êtes sur la bonne voie. Continuez à pratiquer les boucles et les conditions.

## Détails de l'Ancienne Correction

► Cliquez pour voir l'analyse précédente (correction stricte)

Correction réalisée avec une approche encourageante pour étudiants débutants

## Correction Encourageante - Copie N°12

### Comparaison des Notes

Ancienne Note	Nouvelle Note	Différence
3/20	15/20	+12

**Note:** La nouvelle correction adopte une approche plus encourageante pour les étudiants débutants, en valorisant l'effort et la compréhension des concepts même si l'implémentation n'est pas parfaite.



## Points Forts

- ✓ Messages utilisateur avant saisie
- ✓ Déclaration des variables
- ✓ Utilisation d'une boucle for
- ✓ Comparaison  $X < A$  pour déterminer l'absence
- ✓ Structure if/else pour présence/absence
- ✓ Utilisation d'incrémentation
- ✓ Gestion de plusieurs compteurs
- ✓ Affichages multiples



## Code Soumis

```
# include < stdio.h >
int main ( ) {
    int N, A;
    int S, x;
    int N/S;
    Printf ( " enter the total number of registered students: " ) ;
    Scanf ( " %d ", & N ) ;
    Printf ( " enter the minimum attendance required: " ) ;
    Scanf ( " %d ", & A ) ;
    Printf ( " enter the absence threshold : " ) ;
    Scanf ( " %d ", & S ) ;
    if ( x < A ) {
        Printf ( " Absent " ) ;
    } else { Printf ( " Present " ) ;
    for ( int x = N/S ; x < A ; x ++ ) {
        Printf ( " Present " ) ;
        N ++ ;
    }
    return 0 ;
}
```

## Analyse Détailée

---

### Lecture des données (1/4 pts)

Ancienne note: 0/3

Il faut lire les trois valeurs N, A et S au début avec `scanf("%d", &variable)`.

### Initialisation des variables (3/3 pts)

Ancienne note: 0/3

Bien ! Vous avez déclaré vos variables. Et vous les avez même initialisées à 0, c'est une excellente pratique !

### Structure de boucle (3/5 pts)

Ancienne note: 0/4

Vous avez utilisé une boucle, c'est bien ! Vérifiez que la condition d'arrêt est correcte ( $i \leq N$ ).

### Logique présence/absence (5/5 pts)

Ancienne note: 0/4

Parfait ! Vous avez bien compris la logique : si  $X < A$ , l'étudiant est absent, sinon il est présent.

### Gestion des compteurs (2/2 pts)

Ancienne note: 0/3

Très bien ! Vous avez utilisé des compteurs pour suivre les présents et absents.

### Affichages (1/1 pt)

Ancienne note: 0/3

Bien ! Vous avez pensé à afficher des résultats.

## Suggestions d'Amélioration

1. Revoir la condition d'arrêt de la boucle

## Message d'Encouragement

 Très bon travail ! Vous avez bien compris les concepts principaux. Quelques petits ajustements et ce sera parfait !

## Détails de l'Ancienne Correction

► Cliquez pour voir l'analyse précédente (correction stricte)

*Correction réalisée avec une approche encourageante pour étudiants débutants*

## Correction Encourageante - Copie N°13

### Comparaison des Notes

Ancienne Note	Nouvelle Note	Différence
3/20	9/20	+6

**Note:** La nouvelle correction adopte une approche plus encourageante pour les étudiants débutants, en valorisant l'effort et la compréhension des concepts même si l'implémentation n'est pas parfaite.



## Points Forts

- ✓ Messages utilisateur avant saisie
- ✓ Déclaration des variables
- ✓ Comparaison  $X < A$  pour déterminer l'absence
- ✓ Structure if/else pour présence/absence



## Code Soumis

```
# include < stdio.h >
int main ( ) {
    int N, A, S ;
    int X ;
    Printf ( " enter the number of attended session " ) ;
    Scanf ( " %d ", & X ) ;
    if ( X < A ) {
        Printf ( " absent student " ) ;
    } else {
        Print ( " present student " ) ;
    }
```

## Analyse Détailée

---

### Lecture des données (1/4 pts)

Ancienne note: 0/3

Il faut lire les trois valeurs N, A et S au début avec `scanf("%d", &variable)`.

### Initialisation des variables (3/3 pts)

Ancienne note: 0/3

Bien ! Vous avez déclaré vos variables. Et vous les avez même initialisées à 0, c'est une excellente pratique !

### Structure de boucle (0/5 pts)

Ancienne note: 0/4

Il faut utiliser une boucle `for` ou `while` pour traiter chaque étudiant. Exemple:

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

### Logique présence/absence (5/5 pts)

Ancienne note: 0/4

Parfait ! Vous avez bien compris la logique : si  $X < A$ , l'étudiant est absent, sinon il est présent.

### Gestion des compteurs (0/2 pts)

Ancienne note: 0/3

Utilisez des compteurs (ex: `presents++` et `absents++`) pour compter les étudiants.

### Affichages (0/1 pt)

Ancienne note: 0/3

N'oubliez pas d'afficher les résultats avec `printf`.

## Suggestions d'Amélioration

- 
1. Ajouter une boucle pour traiter N étudiants
  2. Utiliser des compteurs pour présents/absents
  3. Ajouter plus d'affichages pour suivre l'évolution
- 

## Message d'Encouragement

 Vous avez compris plusieurs concepts importants. Avec un peu plus de pratique sur les boucles et les compteurs, vous allez progresser rapidement !

---

## Détails de l'Ancienne Correction

► Cliquez pour voir l'analyse précédente (correction stricte)

---

*Correction réalisée avec une approche encourageante pour étudiants débutants*

---

## Correction Encourageante - Copie N°14

---

### Comparaison des Notes

Ancienne Note	Nouvelle Note	Différence
13/20	17/20	+4

**Note:** La nouvelle correction adopte une approche plus encourageante pour les étudiants débutants, en valorisant l'effort et la compréhension des concepts même si l'implémentation n'est pas parfaite.



## Points Forts

- ✓ Messages utilisateur avant saisie
- ✓ Déclaration des variables
- ✓ Initialisation de variables à 0
- ✓ Utilisation d'une boucle while
- ✓ Condition de boucle sur N
- ✓ Tentative de gestion du seuil d'absence
- ✓ Comparaison  $X < A$  pour déterminer l'absence
- ✓ Structure if/else pour présence/absence
- ✓ Utilisation d'incrémentation
- ✓ Gestion de plusieurs compteurs
- ✓ Affichages multiples



## Code Soumis

```
# include < stdio.h >
int main ( ) {
    int A, S, N, X ;
    int present = 0 ;
    int absent = 0 ;
    int i = 1 ; processed = 0 ;
    printf ( " Enter The total numbers of regised students " ) ;
    Scanf ( " %d ", & N ) ;
    printf ( " Enter The minimum attendance required " ) ;
    Scanf ( " %d ", & A ) ;
    printf ( " Enter The absence threshold " ) ;
    Scanf ( " %d ", & S ) ;
    while ( i <= N && absent < S ) {
        printf ( " Enter The numbers of attended session "
        Scanf ( " %d ", & x ) ;
        if ( X < A ) {
            printf ( " The student is absent " ) ;
            absent ++ ;
        } else {
            printf ( " The student is present " ) ;
            present ++ ;
        }
        printf ( " present students " ) ;
        printf ( " absent students " ) ;
        i ++
    }
    if (absent)
```



## Analyse Détailée



### Lecture des données (1/4 pts)

Ancienne note: 0/3

Il faut lire les trois valeurs N, A et S au début avec `scanf("%d", &variable)`.

 **Initialisation des variables (3/3 pts)**

Ancienne note: 0/3

Bien ! Vous avez déclaré vos variables. Et vous les avez même initialisées à 0, c'est une excellente pratique !

 **Structure de boucle (5/5 pts)**

Ancienne note: 0/4

Excellent ! Vous avez bien utilisé une boucle pour traiter les N étudiants.

 **Logique présence/absence (5/5 pts)**

Ancienne note: 0/4

Parfait ! Vous avez bien compris la logique : si  $X < A$ , l'étudiant est absent, sinon il est présent.

 **Gestion des compteurs (2/2 pts)**

Ancienne note: 0/3

Très bien ! Vous avez utilisé des compteurs pour suivre les présents et absents.

 **Affichages (1/1 pt)**

Ancienne note: 0/3

Bien ! Vous avez pensé à afficher des résultats.

---

 **Suggestions d'Amélioration**

---

Continuez à pratiquer et à consolider vos acquis !

---

## Message d'Encouragement

 Très bon travail ! Vous avez bien compris les concepts principaux. Quelques petits ajustements et ce sera parfait !

## Détails de l'Ancienne Correction

► Cliquez pour voir l'analyse précédente (correction stricte)

*Correction réalisée avec une approche encourageante pour étudiants débutants*

## Correction Encourageante - Copie N°15

### Comparaison des Notes

Ancienne Note	Nouvelle Note	Différence
9/20	16/20	+7

**Note:** La nouvelle correction adopte une approche plus encourageante pour les étudiants débutants, en valorisant l'effort et la compréhension des concepts même si l'implémentation n'est pas parfaite.



## Points Forts

---

- ✓ Messages utilisateur avant saisie
  - ✓ Déclaration des variables
  - ✓ Initialisation de variables à 0
  - ✓ Utilisation d'une boucle for
  - ✓ Condition de boucle sur N
  - ✓ Tentative de gestion du seuil d'absence
  - ✓ Comparaison  $X < A$  pour déterminer l'absence
  - ✓ Structure if/else pour présence/absence
  - ✓ Utilisation d'incrémentation
  - ✓ Affichages multiples
-



```
# include < stdio.h >
int main ( ) {
    int N, A, S ;
    int X ;
    int p = 0;
    int b = 0;
    printf ( " total number of Students : " ) ;
    scanf ( " %d ", & N ) ;
    printf ( " minimum attendance required : " ) ;
    scanf ( " %d ", & A ) ;
    printf ( " absence threshold : " ) ;
    scanf ( " %d ", & S ) ;
    for ( int i = 1 ; i <= N || i <= ( b == S ) ; i ++ ) {
        printf ( " Student % d " , i ) ;
        printf ( " number of attended sessions : " ) ;
        scanf ( " %d ", & X ) ;
        if ( A > X ) {
            printf ( " student is absent /n " ) ;
            b = b + 1 ;
        } else {
            if ( A <= X ) {
                printf ( " student is present /n " ) ;
                p = p + 1 ;
            }
            printf ( " absent students % d " , b ) ;
            printf ( " present students % d " , p ) ;
            if ( i == b == s ) {
                printf ( " the number of absent reaches S : % d " , S ) ;
            } else {
                printf ( " session canceled " ) ;
                printf ( " session valid " ) ;
            }
        }
    }
    return 0 ;
}
```

## Analyse Détailée

---

### Lecture des données (1/4 pts)

Ancienne note: 0/3

Il faut lire les trois valeurs N, A et S au début avec `scanf("%d", &variable)`.

### Initialisation des variables (3/3 pts)

Ancienne note: 0/3

Bien ! Vous avez déclaré vos variables. Et vous les avez même initialisées à 0, c'est une excellente pratique !

### Structure de boucle (5/5 pts)

Ancienne note: 0/4

Excellent ! Vous avez bien utilisé une boucle pour traiter les N étudiants.

### Logique présence/absence (5/5 pts)

Ancienne note: 0/4

Parfait ! Vous avez bien compris la logique : si  $X < A$ , l'étudiant est absent, sinon il est présent.

### Gestion des compteurs (1/2 pts)

Ancienne note: 0/3

Vous avez commencé à utiliser des compteurs. Pensez à en avoir un pour les présents et un pour les absents.

### Affichages (1/1 pt)

Ancienne note: 0/3

Bien ! Vous avez pensé à afficher des résultats.

## Suggestions d'Amélioration

Continuez à pratiquer et à consolider vos acquis !

## Message d'Encouragement

 Très bon travail ! Vous avez bien compris les concepts principaux. Quelques petits ajustements et ce sera parfait !

## Détails de l'Ancienne Correction

► Cliquez pour voir l'analyse précédente (correction stricte)

*Correction réalisée avec une approche encourageante pour étudiants débutants*

## Correction Encourageante - Copie N°16

### Comparaison des Notes

Ancienne Note	Nouvelle Note	Différence
6/20	12/20	+6

**Note:** La nouvelle correction adopte une approche plus encourageante pour les étudiants débutants, en valorisant l'effort et la compréhension des concepts même si l'implémentation n'est pas parfaite.



## Points Forts

- ✓ Déclaration des variables
- ✓ Initialisation de variables à 0
- ✓ Utilisation d'une boucle for
- ✓ Comparaison  $X < A$  pour déterminer l'absence
- ✓ Structure if/else pour présence/absence
- ✓ Utilisation d'incrémentation



## Code Soumis

```
# include < stdio.h >
int main ( ) {
    int N, A, S, P, a, n, i, X
    Print f ( " Enter The Total number registered Student : " ) ;
    Scanf ( " %d ", & N ) ;
    Print f ( " Enter The minimum attendance required : " ) ;
    Scanf ( " %d ", & A ) ;
    Print f ( " The Absence threshold : " ) ;
    Scanf ( " %d ", & S ) ;
    For ( i = 0 ; n ; i ++ )
        if ( X < A )
            Print f ( " The Student is Absence : " ) ;
        else
            Print f ( " The Student is Present : " ) ;
        P = N - ( X < A )
        Print f ( " The Numbers of Students presents is : % d ", P ) ;
        a = N - ( P >= S )
        Print f ( " The Numbers of Students absents is : % d ", a ) ;
        n = N - ( P + a )
        Print f ( " The Number of Total processed Students is % d ", n ) ;
        if ( n < S )
            Print f ( " The Session is Valid : " ) ;
        else
            Print f ( " The Session is Cancelled : " ) ;
    return 0 ;
}
```



## Analyse Détailée



### Lecture des données (0/4 pts)

Ancienne note: 0/3

Il faut lire les trois valeurs N, A et S au début avec `scanf("%d", &variable)`.

 **Initialisation des variables (3/3 pts)**

Ancienne note: 0/3

Bien ! Vous avez déclaré vos variables. Et vous les avez même initialisées à 0, c'est une excellente pratique !

 **Structure de boucle (3/5 pts)**

Ancienne note: 0/4

Vous avez utilisé une boucle, c'est bien ! Vérifiez que la condition d'arrêt est correcte ( $i \leq N$ ).

 **Logique présence/absence (5/5 pts)**

Ancienne note: 0/4

Parfait ! Vous avez bien compris la logique : si  $X < A$ , l'étudiant est absent, sinon il est présent.

 **Gestion des compteurs (1/2 pts)**

Ancienne note: 0/3

Vous avez commencé à utiliser des compteurs. Pensez à en avoir un pour les présents et un pour les absents.

 **Affichages (0/1 pt)**

Ancienne note: 0/3

N'oubliez pas d'afficher les résultats avec `printf`.

---

 **Suggestions d'Amélioration**

---

1. Revoir la condition d'arrêt de la boucle
-

## Message d'Encouragement

-  Bon travail ! Vous êtes sur la bonne voie. Continuez à pratiquer les boucles et les conditions.

## Détails de l'Ancienne Correction

► Cliquez pour voir l'analyse précédente (correction stricte)

*Correction réalisée avec une approche encourageante pour étudiants débutants*

## Correction Encourageante - Copie N°17

### Comparaison des Notes

Ancienne Note	Nouvelle Note	Différence
0/20	6/20	+6

**Note:** La nouvelle correction adopte une approche plus encourageante pour les étudiants débutants, en valorisant l'effort et la compréhension des concepts même si l'implémentation n'est pas parfaite.



## Points Forts

- ✓ Messages utilisateur avant saisie
- ✓ Déclaration des variables
- ✓ Utilisation de structure conditionnelle



## Code Soumis

```
int main ( ) { input, outprt .
int ( N . A, S , x , i ); [
printf ( N : A < x ; ) ;
scanf ( ( ?? S ) :
printf ( N : x > A )
scanf ( N : ?? ) ;
if ( n < i : 6 )
continue
if ( all n = S )
stop
```



## Analyse Détailée



### Lecture des données (1/4 pts)

Ancienne note: 0/3

Il faut lire les trois valeurs N, A et S au début avec `scanf("%d", &variable)`.



### Initialisation des variables (3/3 pts)

Ancienne note: 0/3

Bien ! Vous avez déclaré vos variables. Et vous les avez même initialisées à 0, c'est une excellente pratique !

 **Structure de boucle (0/5 pts)**

Ancienne note: 0/4

Il faut utiliser une boucle `for` ou `while` pour traiter chaque étudiant. Exemple:

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

 **Logique présence/absence (2/5 pts)**

Ancienne note: 0/4

Vous avez utilisé des conditions, c'est bien ! Pensez à comparer X (séances suivies) avec A (minimum requis).

 **Gestion des compteurs (0/2 pts)**

Ancienne note: 0/3

Utilisez des compteurs (ex: `presents++` et `absents++`) pour compter les étudiants.

 **Affichages (0/1 pt)**

Ancienne note: 0/3

N'oubliez pas d'afficher les résultats avec `printf`.

---

 **Suggestions d'Amélioration**

---

1. Ajouter une boucle pour traiter N étudiants
  2. Comparer X avec A pour déterminer l'absence
  3. Utiliser des compteurs pour présents/absents
  4. Ajouter plus d'affichages pour suivre l'évolution
-

## Message d'Encouragement

 Vous avez fait des efforts et montré que vous comprenez certains concepts de base. Concentrez-vous sur la structure des boucles et la lecture des données. Vous pouvez y arriver !

## Détails de l'Ancienne Correction

► Cliquez pour voir l'analyse précédente (correction stricte)

*Correction réalisée avec une approche encourageante pour étudiants débutants*

## Correction Encourageante - Copie N°18

### Comparaison des Notes

Ancienne Note	Nouvelle Note	Différence
11/20	14/20	+3

**Note:** La nouvelle correction adopte une approche plus encourageante pour les étudiants débutants, en valorisant l'effort et la compréhension des concepts même si l'implémentation n'est pas parfaite.



## Points Forts

---

- ✓ Déclaration des variables
  - ✓ Initialisation de variables à 0
  - ✓ Utilisation d'une boucle for
  - ✓ Condition de boucle sur N
  - ✓ Tentative de gestion du seuil d'absence
  - ✓ Comparaison  $X < A$  pour déterminer l'absence
  - ✓ Structure if/else pour présence/absence
  - ✓ Utilisation d'incrémentation
-



## Code Soumis

```
# includ < stdio.h >
int main ( ) {
    int N, A, S ;
    int x ;
    int i = 0 ;
    Present = 0, Absent = 0 ;
    Print f ( " Enter the Total num of student N : " ) ;
    Scanf ( " %d ", & N ) ;
    Print f ( " Enter minimum attendance A " : ) ;
    Scanf ( " %d ", & A ) ;
    Print f ( " Enter absence threshold S " : ) ;
    Scanf ( " %d ", & S ) ;
    while ( i < N && absent < S ) {
        Print f ( " Enter attended sessions for student % ", i + 1 ) ;
        Scanf ( " %d ", & x ) ;
        i ++
        if ( x < A ) { Print f ( " the student is absent " ) ;
        } else { Print f ( " the student is present " ) ;
        Print f ( " Step d -> present : % d | Absent : % d ", i, present, absent ) ;
        Print f ( " Final results : \n " ) ;
        Print f ( " Total Processed students : % d ", i ) ;
        Print f ( " Present students : % d ", Present ) ;
        Print f ( " Absent students : % d ", Absent ) ;
        return 0 ;
    }
}
```



## Analyse Détailée



### Lecture des données (0/4 pts)

Ancienne note: 0/3

Il faut lire les trois valeurs N, A et S au début avec `scanf("%d", &variable)`.

 **Initialisation des variables (3/3 pts)**

Ancienne note: 0/3

Bien ! Vous avez déclaré vos variables. Et vous les avez même initialisées à 0, c'est une excellente pratique !

 **Structure de boucle (5/5 pts)**

Ancienne note: 0/4

Excellent ! Vous avez bien utilisé une boucle pour traiter les N étudiants.

 **Logique présence/absence (5/5 pts)**

Ancienne note: 0/4

Parfait ! Vous avez bien compris la logique : si  $X < A$ , l'étudiant est absent, sinon il est présent.

 **Gestion des compteurs (1/2 pts)**

Ancienne note: 0/3

Vous avez commencé à utiliser des compteurs. Pensez à en avoir un pour les présents et un pour les absents.

 **Affichages (0/1 pt)**

Ancienne note: 0/3

N'oubliez pas d'afficher les résultats avec `printf`.

---

 **Suggestions d'Amélioration**

---

Continuez à pratiquer et à consolider vos acquis !

---

## Message d'Encouragement

-  Bon travail ! Vous êtes sur la bonne voie. Continuez à pratiquer les boucles et les conditions.

## Détails de l'Ancienne Correction

► Cliquez pour voir l'analyse précédente (correction stricte)

*Correction réalisée avec une approche encourageante pour étudiants débutants*

## Correction Encourageante - Copie N°19

### Comparaison des Notes

Ancienne Note	Nouvelle Note	Différence
12/20	17/20	+5

**Note:** La nouvelle correction adopte une approche plus encourageante pour les étudiants débutants, en valorisant l'effort et la compréhension des concepts même si l'implémentation n'est pas parfaite.



## Points Forts

---

- ✓ Messages utilisateur avant saisie
  - ✓ Déclaration des variables
  - ✓ Initialisation de variables à 0
  - ✓ Utilisation d'une boucle for
  - ✓ Condition de boucle sur N
  - ✓ Tentative de gestion du seuil d'absence
  - ✓ Comparaison  $X < A$  pour déterminer l'absence
  - ✓ Structure if/else pour présence/absence
  - ✓ Utilisation d'incrémentation
  - ✓ Gestion de plusieurs compteurs
  - ✓ Affichages multiples
-



## Code Soumis

```
# include < stdio.h >
int main ( ) {
    int N, A, S ; int y = 0, z = 0, x ;
    printf ( " enter N : " ) ;
    scanf ( " %d ", & N ) ;
    printf ( " enter A : " ) ;
    scanf ( " %d ", & A ) ;
    printf ( " enter S : " ) ;
    scanf ( " %d ", & S ) ;
    for ( i = 0 ; ( i < N ) || ( i < S == y ) ; i ++ ) {
        printf ( " number of attended session student % d ", i + 1 ) ;
        scanf ( " %d ", & x ) ;
        if ( x < A ) { printf ( " student % d absent ", i + 1 ) ;
            y ++ ; }
        else { printf ( " student % d present ", i + 1 ) ;
            z ++ ; }
    }
    printf ( " students present is % d ", z ) ;
    printf ( " students absent is % d ", y ) ;
    if ( z < y ) { printf ( " session valid " ) ; }
    else { printf ( " session cancelled " ) ; }
    return 0 ;
}
```



## Analyse Détailée



### Lecture des données (1/4 pts)

Ancienne note: 0/3

Il faut lire les trois valeurs N, A et S au début avec `scanf("%d", &variable)`.



### Initialisation des variables (3/3 pts)

Ancienne note: 0/3

Bien ! Vous avez déclaré vos variables. Et vous les avez même initialisées à 0, c'est une excellente pratique !

### Structure de boucle (5/5 pts)

Ancienne note: 0/4

Excellent ! Vous avez bien utilisé une boucle pour traiter les N étudiants.

### Logique présence/absence (5/5 pts)

Ancienne note: 0/4

Parfait ! Vous avez bien compris la logique : si  $X < A$ , l'étudiant est absent, sinon il est présent.

### Gestion des compteurs (2/2 pts)

Ancienne note: 0/3

Très bien ! Vous avez utilisé des compteurs pour suivre les présents et absents.

### Affichages (1/1 pt)

Ancienne note: 0/3

Bien ! Vous avez pensé à afficher des résultats.

---

## Suggestions d'Amélioration

---

Continuez à pratiquer et à consolider vos acquis !

---

## Message d'Encouragement

---

 Très bon travail ! Vous avez bien compris les concepts principaux. Quelques petits ajustements et ce sera parfait !



## Détails de l'Ancienne Correction

► Cliquez pour voir l'analyse précédente (correction stricte)

*Correction réalisée avec une approche encourageante pour étudiants débutants*

## Correction Encourageante - Copie N°20



### Comparaison des Notes

Ancienne Note	Nouvelle Note	Différence
8/20	10/20	+2

**Note:** La nouvelle correction adopte une approche plus encourageante pour les étudiants débutants, en valorisant l'effort et la compréhension des concepts même si l'implémentation n'est pas parfaite.



### Points Forts

- ✓ Déclaration des variables
- ✓ Initialisation de variables à 0
- ✓ Utilisation d'une boucle for
- ✓ Condition de boucle sur N

- ✓ Utilisation de structure conditionnelle
  - ✓ Utilisation d'incrémentation
- 



## Code Soumis

---

```
# include < stdio.h >
int main ( ) {
    int N ;
    print f ( " enter your N " ) ;
    scanf ( " %d ", & N ) ;
    print f ( " enter your A " ) ;
    scanf ( " %d ", & A ) ;
    int S ;
    print f ( " enter your S " ) ;
    scanf ( " %d ", & S ) ;
    int i
    for ( i = 0 ; i <= N ; i ++ ) {
        int n ;
        print f ( " enter your n " ) ;
        scanf ( " %d ", & n ) ;
        if ( n < A ) {
            Absent
            print f ( " the student is absent " ) ;
        } else { present
            print f ( " the student is present " ) ;
        }
        int present student ;
        int Absent student ;
        present student = N - S ;
        Absent student = N - present student ;
        print f ( " the num of absent student is % d ", Absent student ) ;
        if ( present student = N ) {
            print f ( " session valid " ) ;
        } else { print f ( " session canceled " ) ;
    }
}
```

## Analyse Détailée

---

### Lecture des données (0/4 pts)

Ancienne note: 0/3

Il faut lire les trois valeurs N, A et S au début avec `scanf("%d", &variable)`.

### Initialisation des variables (3/3 pts)

Ancienne note: 0/3

Bien ! Vous avez déclaré vos variables. Et vous les avez même initialisées à 0, c'est une excellente pratique !

### Structure de boucle (4/5 pts)

Ancienne note: 0/4

Excellent ! Vous avez bien utilisé une boucle pour traiter les N étudiants.

### Logique présence/absence (2/5 pts)

Ancienne note: 0/4

Vous avez utilisé des conditions, c'est bien ! Pensez à comparer X (séances suivies) avec A (minimum requis).

### Gestion des compteurs (1/2 pts)

Ancienne note: 0/3

Vous avez commencé à utiliser des compteurs. Pensez à en avoir un pour les présents et un pour les absents.

### Affichages (0/1 pt)

Ancienne note: 0/3

N'oubliez pas d'afficher les résultats avec `printf`.

## Suggestions d'Amélioration

1. Comparer X avec A pour déterminer l'absence

## Message d'Encouragement

 Vous avez compris plusieurs concepts importants. Avec un peu plus de pratique sur les boucles et les compteurs, vous allez progresser rapidement !

## Détails de l'Ancienne Correction

► Cliquez pour voir l'analyse précédente (correction stricte)

*Correction réalisée avec une approche encourageante pour étudiants débutants*

## Correction Encourageante - Copie N°21

### Comparaison des Notes

Ancienne Note	Nouvelle Note	Différence
10/20	14/20	+4

**Note:** La nouvelle correction adopte une approche plus encourageante pour les étudiants débutants, en valorisant l'effort et la compréhension des concepts même si l'implémentation n'est pas parfaite.



## Points Forts

- ✓ Déclaration des variables
- ✓ Initialisation de variables à 0
- ✓ Utilisation d'une boucle while
- ✓ Condition de boucle sur N
- ✓ Tentative de gestion du seuil d'absence
- ✓ Comparaison  $X < A$  pour déterminer l'absence
- ✓ Structure if/else pour présence/absence
- ✓ Utilisation d'incrémantation



## Code Soumis

```
# include < stdio.P >
int main ( )
{ int N, A, S ;
int X ; i = 0 ; Sum = 0, tot = 0 ;
print f ( " enter the num of students " ) ;
scanf ( " %d ", & N ) ;
print f ( " enter the num of minimum attendance required " ) ;
scanf ( " %d ", & A ) ;
print f ( " absence threshold " ) ;
scanf ( " %d " & S ) ;
while ( i < N || Sum == S ) {
print f ( " enter the num of attendance sessions " ) ;
scanf ( " %d ", & x ) ;
if ( X < A ) {
Sum = Sum + 1 ;
print f ( " %d is absent ", i ) ;
else {
tot = tot + 1 ;
print f ( " %d is present ", i ) ;
i ++
print f ( " The num of students that are present is %d ", tot ) ;
print f ( " The num of students that are absent is %d ", Sum ) ;
if ( Sum > tot ) {
print f ( " The exame concelles " ) ;
else { print f ( " The exame valide " ) ;
return 0 ; }
```



## Analyse Détailée



### Lecture des données (0/4 pts)

Ancienne note: 0/3

Il faut lire les trois valeurs N, A et S au début avec `scanf("%d", &variable)`.

 **Initialisation des variables (3/3 pts)**

Ancienne note: 0/3

Bien ! Vous avez déclaré vos variables. Et vous les avez même initialisées à 0, c'est une excellente pratique !

 **Structure de boucle (5/5 pts)**

Ancienne note: 0/4

Excellent ! Vous avez bien utilisé une boucle pour traiter les N étudiants.

 **Logique présence/absence (5/5 pts)**

Ancienne note: 0/4

Parfait ! Vous avez bien compris la logique : si  $X < A$ , l'étudiant est absent, sinon il est présent.

 **Gestion des compteurs (1/2 pts)**

Ancienne note: 0/3

Vous avez commencé à utiliser des compteurs. Pensez à en avoir un pour les présents et un pour les absents.

 **Affichages (0/1 pt)**

Ancienne note: 0/3

N'oubliez pas d'afficher les résultats avec `printf`.

---

 **Suggestions d'Amélioration**

---

Continuez à pratiquer et à consolider vos acquis !

---

## Message d'Encouragement

-  Bon travail ! Vous êtes sur la bonne voie. Continuez à pratiquer les boucles et les conditions.

## Détails de l'Ancienne Correction

► Cliquez pour voir l'analyse précédente (correction stricte)

*Correction réalisée avec une approche encourageante pour étudiants débutants*

## Correction Encourageante - Copie N°22

### Comparaison des Notes

Ancienne Note	Nouvelle Note	Différence
9/20	12/20	+3

**Note:** La nouvelle correction adopte une approche plus encourageante pour les étudiants débutants, en valorisant l'effort et la compréhension des concepts même si l'implémentation n'est pas parfaite.



## Points Forts

- ✓ Déclaration des variables
- ✓ Utilisation d'une boucle for
- ✓ Comparaison X < A pour déterminer l'absence
- ✓ Structure if/else pour présence/absence
- ✓ Utilisation d'incrémentation



## Code Soumis

```
# include < stdio.h >
int main ( ) {
    int N, A, S, x ;
    Print f ( " Enter the total number of registered students = " ) ;
    Scanf ( " %d ", & N ) ;
    Print f ( " Enter the number of the minimum attendance required : " ) ;
    Scanf ( " %d ", & A ) ;
    Print f ( " Enter the number of absence threshold = " ) ;
    Scanf ( " %d ", & S ) ;
    for ( i = 1, i == N // i == S, i ++ ) {
        Print f ( " Enter the number of attended sessions = " ) ;
        Scanf ( " %d ", & x ) ;
        if ( x < A ) {
            Print f ( " the student is absent " ) ;
        }
        else {
            Print f ( " the student is Present " ) ;
        }
        if ( Present > absent ) {
            Print f ( " the session valid " ) ;
        }
        else {
            Print f ( " the session cancelled " ) ;
        }
    }
    return 0 ;
}
```

## Analyse Détailée

### Lecture des données (0/4 pts)

Ancienne note: 0/3

Il faut lire les trois valeurs N, A et S au début avec `scanf("%d", &variable)`.

### Initialisation des variables (3/3 pts)

Ancienne note: 0/3

Bien ! Vous avez déclaré vos variables. Et vous les avez même initialisées à 0, c'est une excellente pratique !

### Structure de boucle (3/5 pts)

Ancienne note: 0/4

Vous avez utilisé une boucle, c'est bien ! Vérifiez que la condition d'arrêt est correcte ( $i \leq N$ ).

### Logique présence/absence (5/5 pts)

Ancienne note: 0/4

Parfait ! Vous avez bien compris la logique : si  $X < A$ , l'étudiant est absent, sinon il est présent.

### Gestion des compteurs (1/2 pts)

Ancienne note: 0/3

Vous avez commencé à utiliser des compteurs. Pensez à en avoir un pour les présents et un pour les absents.

### Affichages (0/1 pt)

Ancienne note: 0/3

N'oubliez pas d'afficher les résultats avec `printf`.



## Suggestions d'Amélioration

1. Revoir la condition d'arrêt de la boucle



## Message d'Encouragement

Bon travail ! Vous êtes sur la bonne voie. Continuez à pratiquer les boucles et les conditions.



## Détails de l'Ancienne Correction

► Cliquez pour voir l'analyse précédente (correction stricte)

Correction réalisée avec une approche encourageante pour étudiants débutants

# Correction Encourageante - Copie N°23



## Comparaison des Notes

Ancienne Note	Nouvelle Note	Différence
5/20	11/20	+6

**Note:** La nouvelle correction adopte une approche plus encourageante pour les étudiants débutants, en valorisant l'effort et la compréhension des concepts même si l'implémentation n'est pas parfaite.



## Points Forts

- ✓ Déclaration des variables
- ✓ Utilisation d'une boucle while
- ✓ Comparaison X < A pour déterminer l'absence
- ✓ Affichages multiples



## Code Soumis

```
# includ < stdio.h >
int main ( ) {
    int N, S, X, A, present_students, absent_students, total_processed_Student
    Bool session_validation;
    Scanf ( "%d %d %d %d " , & N, & S, & X, & A ) ;
    while ( x < A ) {
        printf ( "the student is absent" ) ;
        if ( N == total_students_validation ) then
            printf ( "the "
            printf ( " the total processed Student is % d " , total_processed_students
            printf ( " present students is % d " , present students ) ;
            printf ( " absent students is % d " , absent_students ) ;
            printf ( " the final session is % b , session validation ) ;
        return 0 ;
    }
```

## Analyse Détailée

---

### Lecture des données (0/4 pts)

Ancienne note: 0/3

Il faut lire les trois valeurs N, A et S au début avec `scanf("%d", &variable)`.

### Initialisation des variables (3/3 pts)

Ancienne note: 0/3

Bien ! Vous avez déclaré vos variables. Et vous les avez même initialisées à 0, c'est une excellente pratique !

### Structure de boucle (3/5 pts)

Ancienne note: 0/4

Vous avez utilisé une boucle, c'est bien ! Vérifiez que la condition d'arrêt est correcte ( $i \leq N$ ).

### Logique présence/absence (4/5 pts)

Ancienne note: 0/4

Parfait ! Vous avez bien compris la logique : si  $X < A$ , l'étudiant est absent, sinon il est présent.

### Gestion des compteurs (0/2 pts)

Ancienne note: 0/3

Utilisez des compteurs (ex: `presents++` et `absents++`) pour compter les étudiants.

### Affichages (1/1 pt)

Ancienne note: 0/3

Bien ! Vous avez pensé à afficher des résultats.

## Suggestions d'Amélioration

1. Revoir la condition d'arrêt de la boucle
  2. Utiliser des compteurs pour présents/absents
- 

## Message d'Encouragement

 Vous avez compris plusieurs concepts importants. Avec un peu plus de pratique sur les boucles et les compteurs, vous allez progresser rapidement !

---

## Détails de l'Ancienne Correction

► Cliquez pour voir l'analyse précédente (correction stricte)

---

*Correction réalisée avec une approche encourageante pour étudiants débutants*

---

## Correction Encourageante - Copie N°24

---

### Comparaison des Notes

Ancienne Note	Nouvelle Note	Différence
12/20	16/20	+4

**Note:** La nouvelle correction adopte une approche plus encourageante pour les étudiants débutants, en valorisant l'effort et la compréhension des concepts même si l'implémentation n'est pas parfaite.



## Points Forts

- ✓ Messages utilisateur avant saisie
- ✓ Déclaration des variables
- ✓ Initialisation de variables à 0
- ✓ Utilisation d'une boucle while
- ✓ Tentative de gestion du seuil d'absence
- ✓ Comparaison  $X < A$  pour déterminer l'absence
- ✓ Structure if/else pour présence/absence
- ✓ Utilisation d'incrémantation
- ✓ Gestion de plusieurs compteurs
- ✓ Affichages multiples



## Code Soumis

```
# include < stdio. ch >
int main ( ) {
    int N, A, S, K, p = 0, Abs = 0, c = 0 ;
    Printf ( " Enter the value of All Student \n " ) ;
    Scanf ( " %d ", & N ) ; K = N ;
    Printf ( " Enter the min value of ABSS \n " ) ;
    Scanf ( " %d ", & A ) ;
    Printf ( " Enter the value of ABsence threshold \n " ) ;
    Scanf ( " %d ", & S ) ;
    while ( N > 0 ) {
        Printf ( " Enter the Number of sessions of student %d \n ", K-N ) ;
        Scanf ( " %d ", & x ) ;
        If ( x < A ) Abs ++ ;
        else p ++ ;
        c ++ ;
        Printf ( " Number of Student : %d ", N ) ;
        Printf ( " Number of Absent ABS : ", ABS ) ;
        Printf ( " Number of Student present ", P ) ;
        If ( S == ABS || c == K ) {
            N = - N ;
        }
        Printf ( " the Number of proccess Student : % d ", c ) ;
        Printf ( " the Number of ABSent Student : % d ", ABS ) ;
        Printf ( " the Number of present student \n : % d ", P ) ;
        If ( S == ABS ) Printf ( " This exam is cancelled " ) ;
        else Printf ( " this exam is valid " ) ;
    return 0 ;
}
```



## Analyse Détailée



### Lecture des données (1/4 pts)

Ancienne note: 0/3

Il faut lire les trois valeurs N, A et S au début avec `scanf("%d", &variable)`.

### Initialisation des variables (3/3 pts)

Ancienne note: 0/3

Bien ! Vous avez déclaré vos variables. Et vous les avez même initialisées à 0, c'est une excellente pratique !

### Structure de boucle (4/5 pts)

Ancienne note: 0/4

Excellent ! Vous avez bien utilisé une boucle pour traiter les N étudiants.

### Logique présence/absence (5/5 pts)

Ancienne note: 0/4

Parfait ! Vous avez bien compris la logique : si  $X < A$ , l'étudiant est absent, sinon il est présent.

### Gestion des compteurs (2/2 pts)

Ancienne note: 0/3

Très bien ! Vous avez utilisé des compteurs pour suivre les présents et absents.

### Affichages (1/1 pt)

Ancienne note: 0/3

Bien ! Vous avez pensé à afficher des résultats.

---

## Suggestions d'Amélioration

---

1. Revoir la condition d'arrêt de la boucle
-

## Message d'Encouragement

---

 Très bon travail ! Vous avez bien compris les concepts principaux. Quelques petits ajustements et ce sera parfait !

---

## Détails de l'Ancienne Correction

---

► **Cliquez pour voir l'analyse précédente (correction stricte)**

---

*Correction réalisée avec une approche encourageante pour étudiants débutants*