

Copy number :

1

تعليمات إلزامية : كتابة البرنامج كاملاً داخل main | استعمال حلقة واحدة فقط break / continue | يمنع استعمال المصفوفات، الدوال،

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

```
# include < stdio.h >
# include < stdio.h >
int main () {
    const N;
    int A, 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);
    int i, x;
    for (i = 0 ; i <= x ; i++) {
        printf ("Enter x");
        if (x < A) {
            printf ("the student is absent");
        } else {
            printf ("the student is present");
        }
        scanf ("%d", &x);
    }
    printf ("%d %d", number of present and absent students);
    if (number of Absent student > S) {
        printf ("Session cancelled");
    } else {
        printf ("Session valid");
    }
    return 0;
}
```

Copy 1

```
#include <stdio.h>
int main()
{
    const N;
    int A, 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);
    int i, x;
    for (i = 0; i <= x; i++)
    {
        printf("Enter x");
        if (x <A)
        {
            printf("the student is absent");
        }
        else
        {
            printf("the student is present");
        }
        scanf("%d", &x);
    }

    printf("%d %d", number of present and absent students);
    if (number of Absent student> S)
    {
        printf("Session cancelled");
    }
    else
    {
        printf("Session valid");
    }
    return 0;
}
```

COPY NUMBER: 1

Analyse :

- **Contraintes** : Respectées (une seule boucle, stdio.h, pas de tableau).

Algorithmique :

- Erreur syntaxe déclaration : `const N;` au lieu de `int N;` ou `const int N = ...;`.
- Boucle `for (i = 0; i <= x; i++)` : La condition d'arrêt dépend de `x` (variable non initialisée ou saisie dans la boucle), et non de `N` ou `S`. Logique de boucle incorrecte.
- Lecture de `x` à l'intérieur de la boucle deux fois (`printf` puis `scanf`).
- Le `printf` final utilise une variable `number of present and absent students` qui n'existe pas (pseudo-code non valide en C).
- Condition finale `if (number of Absent student > S)` utilise aussi du pseudo-code.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	2 / 3	<code>scanf</code> présents mais déclaration <code>const N</code> invalide.
Initialisation	1 / 3	Variables non initialisées correctement (notamment compteurs).
Condition boucle	0 / 4	<code>i <= x</code> est insensé ici.
Logique prés./abs.	2 / 4	<code>if (x < A)</code> correct, mais dans une boucle mal structurée.
Compteurs	0 / 3	Aucun compteur incrémenté.
Affichages inter.	1 / 2	Messages présents mais pas de compteurs affichés.
Affichage final	0 / 1	Utilisation de pseudo-code invalide (<code>number of...</code>).

NOTE FINALE : 06 / 20

Feedback :

- **Points forts** : Structure globale du programme (main, includes, structure if/else).
- **Points faibles** : Confusion entre code C et pseudo-code (noms de variables). Logique de la boucle `for` à revoir complètement.

- **Appréciation globale : Fragile.** Les bases de la syntaxe sont à consolider. Attention à ne pas inventer de noms de variables avec des espaces.
-

Copy number :

2

تعليمات إلزامية : كتابة البرنامج كاملاً داخل main | استعمال حلقة واحدة فقط | يمنع استعمال المصفوفات، الدوال، break / continue

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, M;
    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);
    printf("The number of attended Sessions");
    scanf("%d", &X);
    for (int i = 1; i <= N || i <= S; ++i) {
        if (x < A) {
            printf("absence Students");
        } else {
            printf("Present Students");
        }
        if (i >= S) {
            printf("Session Failed");
        } else {
            printf("Session Completed");
        }
        printf("\n");
    }
    return 0;
}
```

Copy 2

```
#include <stdio.h>
int main()
{
    int N, A, S, X, M;
    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);
    printf("The number of attended sessions");
    scanf("%d", &X);
    for (int i = 1; i <= N || i <= S; i++)
    {
        if (X < A)
        {
            printf("absence students");
        }
        else
        {
            printf("present students");
        }
    }

    if (M>= S)
    {
        printf("Session Valid");
    }
    else
    {
        printf("Session Cancelled");
    }

    printf("%d", M);
    return 0;
}
```

COPY NUMBER: 2

Analyse :

- **Contraintes** : Respectées.

Algorithmique :

- Déclarations correctes.
- Boucle `for` : Condition `i <= N || i <= S` est étrange (`||` au lieu de `&&` probablement, et comparaison `i <= S` incorrecte car `S` est un seuil d'absents, pas d'itérations). Indexation `i` commence à 1.
- Lecture de `x` : Fait **avant** la boucle (donc une seule fois pour tous ?). Devrait être *dans* la boucle.
- Prise de décision : Affiche "absence/present" mais n'incrémente pas de compteurs explicites pour `absent` ou `present`.
- Condition finale : Utilise `M` (non initialisé) comparé à `S`. `M` semble être utilisé comme compteur mais n'est jamais incrémenté.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initialisation	1 / 3	<code>M</code> non initialisé.
Condition boucle	1 / 4	Utilisation de <code> </code> rend la boucle potentiellement infinie ou incorrecte. <code>i <= S</code> n'a pas de sens ici.
Logique prés./abs.	2 / 4	Test <code>X < A</code> correct, mais <code>X</code> n'est pas lu à chaque itération.
Compteurs	0 / 3	Pas de mise à jour des compteurs.
Affichages inter.	1 / 2	Feedback textuel présent, mais pas de données chiffrées.
Affichage final	0 / 1	Basé sur une variable non initialisée <code>M</code> .

NOTE FINALE : 08 / 20

Feedback :

- **Points forts** : Saisie des paramètres correcte.

- **Points faibles** : Erreur logique majeure : `scanf` hors de la boucle. Gestion des compteurs inexisteante. Confusion sur la condition d'arrêt (OU logique vs ET logique).
 - **Appréciation globale** : **Insuffisant**. La logique répétitive n'est pas acquise.
-

Copy number : 3

تعليمات إلزامية : كتابة البرنامج كاملاً داخل main | استعمال حلقـة واحدة فقط | يمنع استعمال المصفوفات، الدوال، break / continue

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

```
int main();
    int N, A, S, X;
    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);
    read the number of attended sessions x;
    if x < A;
        for ("the student is considered absent");
    if x >= A;
        for ("the student is present");
        if x < A < n = A;
            for ("absent students");
        if n >= A < x < A;
            for ("present students");
        final status
        Session Valid;
    return 0;
```

Copy 3

```
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);
    // read the number of attended sessions x;
    if (X <A)
    {
        printf("the student is considered absent");
    }

    if (X>= A)
    {
        printf("the student is present");
    }

    if (X <A)
    {
        printf("absent students");
    }

    if (X>= A)
    {
        printf("present students");
    }

    // final status printf("Session Valid");
    return 0;
}
```

COPY NUMBER: 3

Analyse :

- **Contraintes** : Manque `#include <stdio.h>`.

Algorithmique :

- Pas de boucle ! Le code s'exécute séquentiellement une seule fois (lit `N`, `A`, `S`, puis teste une fois).
- Commentaire `// read the number of attended sessions x;` mais pas de `scanf` pour `x` avant son utilisation.
- Logique conditionnelle dupliquée.
- Pas de compteurs, pas de gestion de `N` étudiants.

Notation :

Critère	Points	Commentaire
Lecture <code>N</code> , <code>A</code> , <code>S</code>	3 / 3	Correct.
Initialisation	1 / 3	Variables déclarées mais <code>x</code> utilisé sans lecture (valeur indéterminée).
Condition boucle	0 / 4	Aucune boucle présente.
Logique prés./abs.	1 / 4	Structure <code>if</code> présente mais sur une variable non initialisée.
Compteurs	0 / 3	Absents.
Affichages inter.	0 / 2	Inexistants.
Affichage final	0 / 1	Affiche toujours "Session Valid".

NOTE FINALE : 05 / 20

Feedback :

- **Points forts** : Sait lire des entiers.
- **Points faibles** : L'énoncé demandait une simulation pour `N` étudiants. Absence totale de structure itérative. Code incomplet.
- **Appréciation globale** : **Très Insuffisant**. Hors sujet sur l'aspect algorithmique (boucles).

Copy number :

٤

تعليمات إلزامية : كتابة البرنامج كاملاً داخل main | استعمال حلقة واحدة فقط | يمنع استعمال المصفوفات، الدوال، break / continue

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

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

Copy 4

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

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

return 0;
}
```

COPY NUMBER: 4

Analyse :

- **Contraintes** : Manque `#include <stdio.h>`.

Algorithmique :

- Lecture N, A, S correcte.

Boucle `for (i = 0; i < N; i++)` correcte pour itérer N fois. **MAIS** ne gère pas l'arrêt anticipé si `B >= S`.

• Note: Il y a un commentaire `// break // S`, l'étudiant savait qu'il fallait arrêter mais ne l'a pas codé.

- `scanf("%d", &x)` dans la boucle : Correct.

- Compteurs B (absents ?) et P (présents) incrémentés : Correct. **Cependant**, B et P ne sont **pas initialisés** à 0 ! (Comportement indéfini).

- Logique `if (x < A)` correcte.

- Affichage final cohérent.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initialisation	1 / 3	B et P non initialisés à 0.
Condition boucle	2 / 4	Boucle sur N correcte, mais condition d'arrêt sur S manquante dans le <code>for</code> .
Logique prés./abs.	4 / 4	Correcte.
Compteurs	3 / 3	Incrémantation correcte (malgré le défaut d'init).
Affichages inter.	1 / 2	Affiche le statut mais pas les compteurs courants à chaque étape.
Affichage final	1 / 1	Correct.

NOTE FINALE : 15 / 20

Feedback :

- **Points forts** : Bonne structure de boucle, lecture des entrées au bon endroit.
- **Points faibles** : **Oubli critique de l'initialisation** (`int R, P = 0;`). Condition d'arrêt sur le seuil d'absence manquante (le commentaire ne suffit pas).

- **Appréciation globale : Moyen / Bon.** Algorithme quasi-fonctionnel, erreurs d'inattention coûteuses.
-

Copy number :

5

تعليمات إلزامية : كتابة البرنامج كاملاً داخل main | استعمال حلة واحدة فقط | يمنع استعمال المصفوفات، الدوال، break / continue

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

```
int main() {
    int N, S, A, X, i;
    printf ("enter the total num of registered students");
    scanf ("%d", &N);
    for (i=1; i <=N; i++) {
        printf ("enter A in enter m");
        scanf ("%d %d", &A, &X);
        if (X < A) {
            printf ("the student is absent");
        } else {
            printf ("the student is present");
        }
        printf ("enters");
        scanf ("%d", &S);
        if (N >= S) {
            printf ("Session Valid");
        } else {
            printf ("Session cancelled");
        }
    }
    return 0;
}
```

Copy 5

```
int main()
{
    int N, S, A, x, i;
    printf("enter the total num of registered students");
    scanf("%d", &N);
    for (i = 1; i <= N; i++)
    {
        printf("enter A and enter x");
        scanf("%d %d", &A, &x);
        if (x < A)
        {
            printf("the student is absent");
        }
        else
        {
            printf("the student is present");
        }
    }

    printf("enters");
    scanf("%d", &S);
    if (N>= S)
    {
        printf("Session Valid");
    }
    else
    {
        printf("Session cancelled");
    }
}

return 0;
}
```

COPY NUMBER: 5

Analyse :

- **Contraintes** : Manque #include.

Algorithmique :

- Lecture de N.
- Boucle `for (i = 1; i <= N; i++)` : Correcte pour N itérations.
- Lecture `scanf ("%d %d", &A, &x)` **DANS** la boucle : Erreur majeure. A (seuil) ne doit être lu qu'une seule fois au début. Ici l'utilisateur doit le resaisir à chaque étudiant.
- Lecture de S **APRÈS** la boucle : Trop tard. Impossible d'arrêter la simulation si le seuil est atteint pendant.
- Logique `if (x < A)` correcte.
- Pas de compteurs. Affiche juste le statut textuel.
- Condition finale `if (N >= S)` : Compare le nombre total d'étudiants au seuil ? Sens douteux.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	1 / 3	Erreur de placement (A dans la boucle, S après).
Initialisation	3 / 3	Variables déclarées (<code>int</code> par défaut 0 non garanti mais ici pas de compteurs utilisés).
Condition boucle	2 / 4	Boucle sur N correcte, arrêt sur S impossible.
Logique prés./abs.	4 / 4	Comparaison correcte.
Compteurs	0 / 3	Inexistants.
Affichages inter.	1 / 2	Texte uniquement.
Affichage final	0 / 1	Logique incorrecte (<code>N >= S</code>).

NOTE FINALE : 11 / 20

Feedback :

- **Points forts** : Syntaxe de base correcte.
- **Points faibles** : Mauvaise compréhension de l'ordre des instructions (saisie des paramètres vs traitement). Gestion de l'arrêt anticipé manquante.
- **Appréciation globale : Moyen.**

Copy number :

6

تعليمات إلزامية : كتابة البرنامج كاملاً داخل main | استعمال حلقة واحدة فقط | يمنع استعمال المصفوفات، الدوال، break / continue

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, i = 0;
    printf("enter the N:");
    scanf("%d", &N);
    printf("enter the A:");
    scanf("%d", &A);
    printf("enter the S:");
    scanf("%d", &S);
    printf("enter the x:");
    scanf("%d", &x);
    while (i < N) {
        if (x < A) {
            printf("the student is considered absent");
        } else {
            printf("the student is considered present");
        }
        if (absent < S) {
            printf(" is valid");
        }
    }
}
```

Copy 6

```
#include <stdio.h>
int main()
{
    int N, A, S, x, i, m;
    printf("enter the N :");
    scanf("%d", &N);
    printf("enter the A :");
    scanf("%d", &A);
    printf("enter the s :");
    scanf("%d", &S);
    printf("enter the x :");
    scanf("%d", &x);
    for (i = 1; x <= N || x <= S; i++)
    {
        if (x < A)
        {
            printf("the student is considered absent");
        }
        else
        {
            printf("the student is considered present");
        }

        if (absent < S)
        {
            printf("is valid");
        }
        else
        {
            printf("is cancelled");
        }
    }
    return 0;
}
```

COPY NUMBER: 6

Analyse :

Algorithmique :

- Lectures correctes.

Boucle for (*i* = 1; *x* <= N || *x* <= S; *i*++) :

- Condition conditionnée par *x* (variable de saisie courante ?) au lieu de compteurs.
- Mélange || et variables non pertinentes.

- Lecture de *x* avant la boucle ? Non, il y a un scanf avant, mais pas **dans** la boucle (boucle infinie avec la même valeur de *x*, ou arrêt immédiat).
- Utilisation de absent dans le if final, mais variable non déclarée ni calculée.
- return 0 **dans** la boucle, ce qui arrête le programme dès la première itération.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initialisation	2 / 3	Correct.
Condition boucle	0 / 4	Condition illogique et scanf manquant dans la boucle (bloqué ou infini).
Logique prés./abs.	2 / 4	Test présent, mais ne s'exécute pas correctement (pas de scanf interne).
Compteurs	0 / 3	Non implémentés.
Affichages inter.	1 / 2	Texte seul.
Affichage final	0 / 1	Variable absent non définie. Retour prématuré.

NOTE FINALE : 08 / 20

Feedback :

- Points forts** : Début correct.
- Points faibles** : Grosses erreurs de logique de contrôle (boucle mal formée, return prématuré, variable non déclarée). Le programme s'arrête immédiatement.
- Appréciation globale : Fragile.**

Copy number : ٧

تعليمات إلزامية : كتابة البرنامج كاملاً داخل main | استعمال حلقة واحدة فقط | يمنع استعمال المصفوفات، الدوال، break / continue

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, N, S, n; // Input data
    printf ("enter the number of total 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 = 1; i <= N; i++) {
        if (i < A) // for each student
            printf ("the student is considered absent");
        else
            printf ("the student is present");
        if (i == N || N == S)
            printf ("simulation stop.");
        if (A < i)
            printf ("the session valid");
        else
            if (A < i)
                printf ("the session cancelled");
    }
    return 0;
}
```

Copy 7

```
#include <stdio.h>
int main()
{
    int A, N, S, n;
    // Input data printf("Enter the number of total 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 = 1; i <= N; i++)
    {
        if (i < A) // for each student
        {
            printf("the student is considered absent");
        }
        else
        {
            printf("the student is present");
        }
    }

    if (i == N || N == S)
    {
        printf("simulation stop.");
    }

    if (A == i)
    {
        printf("the session valid");
    }
    else
    {
        if (A < i)
        {
            printf("the session cancelled");
        }
    }
}

return 0;
}
```

COPY NUMBER: 7

Analyse :

Algorithmique :

- Lecture correcte.

- Boucle for (`int i = 1; i <= N; i++`). Pas de condition d'arrêt sur S.

Test if (`i < A`) : Compare le **numéro de l'étudiant** (`i`) au seuil d'absence (`A`) !

• Erreur fondamentale. Il fallait lire une variable `x` (heures) et la comparer à `A`.

- Pas de lecture de la présence (`scanf`) dans la boucle.

- Conditions finales confuses (`if (i == N || N == S)`).

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initialisation	3 / 3	OK.
Condition boucle	2 / 4	Itère N fois, mais ne lit rien.
Logique prés./abs.	0 / 4	Compare l'indice de boucle au lieu d'une saisie.
Compteurs	0 / 3	Inexistants.
Affichages inter.	1 / 2	Texte basé sur une logique fausse.
Affichage final	0 / 1	Incohérent.

NOTE FINALE : 09 / 20

Feedback :

- **Points forts** : Code propre visuellement.
- **Points faibles** : Confusion totale sur le fonctionnement : on ne simule pas la présence en comparant le numéro de l'étudiant au seuil. Il manque la saisie des données pour chaque étudiant.
- **Appréciation globale** : **Insuffisant**. Le concept de traitement de données est manqué.

Copy number :

8

تعليمات إلزامية : كتابة البرنامج كاملاً داخل main | استعمال حلقة واحدة فقط | يمنع استعمال المصفوفات، الدوال، break / continue

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, i=1, X, Present S, Absent S, Processed S;

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

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

scanf ("%d", &X);

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

- if (X < A) {

printf ("the student is absent");

} else {

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

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

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

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

printf ("stop the simulation !!");

} else

* printf (* Enter the number of present students *);

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

printf ("Session Valid");

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

printf ("Session cancelled");

return 0; }

}

Copy 8

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

        printf("%d is the number of present students", PresentS);
        printf("%d is the number of absent students", AbsentS);
        if (i == N || AbsentS == S)
        {
            printf("stop the simulation !!!");
        }
        else
        {
            // printf("Enter the number of present students");
            // scanf("%d", &PresentS);
            printf("Session valid");
            // scanf("%d", &AbsentS);
            printf("Session cancelled");
        }
    }
    return 0;
}
```

COPY NUMBER: 8

Analyse :

Algorithmique :

- Lecture groupée `scanf("%d %d %d", ...)` correcte.
- Lecture de `x` **avant** la boucle. Ne sera pas mis à jour.
- Boucle `for (i = 1; i <= N; i++)`.
- Corps de boucle : Affiche "absent/present" selon la valeur initiale de `x`.
- Affiche `PresentS` et `AbsentS` mais ces variables ne sont **jamais initialisées ni incrémentées**. Elles contiennent des valeurs poubelles.
- Test d'arrêt : `if (i == N || AbsentS == S)`. Comme `AbsentS` est poubelle, comportement imprévisible.
- `return 0` **dans** la boucle. S'arrête après la 1ère itération.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initialisation	0 / 3	Variables compteurs non initialisées.
Condition boucle	1 / 4	Boucle interrompue par un <code>return</code> prématuré.
Logique prés./abs.	2 / 4	Comparaison correcte mais sur une donnée fixe.
Compteurs	0 / 3	Aucune mise à jour.
Affichages inter.	0 / 2	Affiche des valeurs poubelles.
Affichage final	0 / 1	Code jamais atteint ou incorrect.

NOTE FINALE : 06 / 20

Feedback :

- **Points forts** : Syntaxe `scanf` concise.
- **Points faibles** : Variables non initialisées (grave en C). `scanf` hors boucle. `return` qui tue la boucle. Usage de variables non calculées.
- **Appréciation globale** : Très Fragile. Des lacunes importantes en algorithmie.

Copy number :

9

تعليمات إلزامية : كتابة البرنامج كاملاً داخل main | استعمال حلقة واحدة فقط | يمنع استعمال المصفوفات، الدوال، break / continue

| 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, i = 0, sum = 0;
    printf("enter the total number registered students : ");
    scanf("%d", &n);
    printf("enter the minimum attendance required : ");
    scanf("%d", &a);
    printf("enter the absence threshold ");
    scanf("%d", &s);
    for(i = 0; i <= n; i++)
    {
        scanf("%d", &n);
        if(a > n)
        {
            sum = sum + 1;
            printf("absent students : %d", sum);
        }
        if(a >= n)
        {
            sum += sum;
            printf("present students : %d", sum);
        }
        if(n == n || n == s)
        {
            printf("stop in");
        }
        if(a > s)
        {
            printf("Session Cancelled");
        }
        else if(s > a)
        {
            printf("Session Valid");
        }
    }
    return 0;
}
```

Copy 9

```
#include <stdio.h>
int main()
{
    int n, a, s, i = 0, x, sum = 0;
    printf("enter the total number registered students");
    scanf("%d", &n);
    printf("enter the minimum attendance required");
    scanf("%d", &a);
    printf("enter the absence threshold");
    scanf("%d", &s);
    for (i = 0; i <n; i++)
    {
        scanf("%d", &x);
        if (a> x)
        {
            sum = sum + 1;
            printf("absent students: %d", sum);
        }

        else if (x>= a)
        {
            sum = sum;
            printf("present students: %d", sum);
        }

        if (i == n || n == s)
        {
            printf("stop in");
        }

        if (a> s)
        {
            printf("Session Cancelled");
        }

        else if (s> a)
        {
            printf("Session Valid");
        }
    }

    return 0;
}
```

COPY NUMBER: 9

Analyse :

Algorithmique :

- Lectures correctes.
- Boucle `for (i = 0; i < n; i++)` : Correcte.
- `scanf("%d", &x)` dans la boucle : Correct.
- Logique : `if (a > x)` (Absent, `sum` non incrémenté ? Ah si `sum = sum + 1`).
- `else if (x >= a)` (Présent, `sum = sum` inutile).
- Utilise `sum` pour compter les absents. Pas de compteur pour présents affiché distinctement.
- Tests d'arrêt dans la boucle : `if (i == n || n == s)` -> Condition `n == s` est constante et ne dépend pas de l'avancement. Devait être `sum == s`.
- Pas de `break` explicite demandé mais condition d'arrêt mal formulée (affiche juste "stop in" sans arrêter).

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initialisation	3 / 3	<code>sum</code> initialisé.
Condition boucle	2 / 4	Boucle sur N ok, mais l'arrêt sur seuil n'est pas effectif (juste un <code>print</code>).
Logique prés./abs.	4 / 4	Correcte.
Compteurs	2 / 3	Compte les absents (<code>sum</code>), mais oublie de gérer/afficher les présents distinctement.
Affichages inter.	1 / 2	Affiche le cumul d'absents.
Affichage final	1 / 1	Logique <code>a > s</code> ? Non, devrait comparer <code>sum</code> et <code>s</code> . Confus.

NOTE FINALE : 16 / 20

Feedback :

- **Points forts** : Code fonctionnel sur la partie itérative de base.
- **Points faibles** : Confusion entre les variables (`a > s` à la fin au lieu de `sum > s`). L'arrêt sur seuil n'arrête pas vraiment la boucle (`break` interdit, il fallait mettre la condition dans le `for`).

- **Appréciation globale : Bon.** L'étudiant a compris le principe général.
-

Copy number :

10

تعليمات إلزامية : كتابة البرنامج كاملاً داخل main | استعمال حلقة واحدة فقط | يمنع استعمال المصفوفات، الدوال، break / continue

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

```
// Pres = the present student  
// Abs = the absent student  
// St = Step = student  
#include <stdio.h>  
int main(){  
    int N, A, S, Pres=0, Abs=0, St=0;  
    scanf ("%d %d %d", &N, &A, &S);  
    while (St != N || Abs != S){  
        St++;  
        int x;  
        scanf ("%d", &x);  
        if (x < A) {Abs++}  
        else  
            Pres++;  
    printf ("Present: %d", Pres);  
    printf ("Absent: %d", Abs);  
    printf ("Step: %d", St);  
    printf ("Students: %d, Present: %d, Absent: %d", St, Pres, Abs);  
    if (Abs < S)  
        printf ("Valid Session");  
    else  
        printf ("Concealed Session");  
    return 0;}
```

Copy 10

```
#include <stdio.h>
int main()
{
    int N, A, S, Pres = 0, Abs = 0, St = 0;
    scanf("%d %d %d", &N, &A, &S);
    while (St != N || Abs != S)
    {
        St++;
        int x;
        scanf("%d", &x);
        if (x <A)
        {
            Abs++;
        }
        else
        {
            Pres++;
        }

        printf("Present: %d", Pres);
        printf("Absent: %d", Abs);
        printf("Step: %d", St);
    }

    printf("Students: %d, Present: %d, Absent: %d", St, Pres, Abs);
    if (Abs <S)
    {
        printf("Valid Session");
    }
    else
    {
        printf("cancelled Session");
    }
}

return 0;
}
```

COPY NUMBER: 10

Analyse :

Algorithmique :

- Lecture groupée OK.
- Boucle while (`St != N || Abs != S`) : L'opérateur devrait être `&&` (Tant que pas fini ET pas seuil atteint). Avec `||`, la boucle continue tant que l'une des conditions est vraie (risque de dépassement).
- Incrémentation `St++`, Lecture `x`, IF/ELSE pour `Abs/Pres`. Tout est correct.
- Affichages intermédiaires complets.
- Affichage final complet : `printf("Students: %d...", St...)`.
- Condition finale : `if (Abs < S)` Correct.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initialisation	3 / 3	Correcte.
Condition boucle	2 / 4	Erreur de logique booléenne : <code> </code> au lieu de <code>&&</code> pour une boucle "Tant que".
Logique prés./abs.	4 / 4	Correcte.
Compteurs	3 / 3	Corrects.
Affichages inter.	2 / 2	Complets.
Affichage final	1 / 1	Correct.

NOTE FINALE : 18 / 20

Feedback :

- **Points forts** : Code très propre, logique claire, affichages conformes.
- **Points faibles** : Attention aux lois de Morgan (while continue tant que condition VRAIE -> (`St < N && Abs < S`)).
- **Appréciation globale : Très Bon.** Excellente copie.

Copy number :

11

تعليمات إلزامية : كتابة البرنامج كاملاً داخل main | استعمال حلقة واحدة فقط | يمنع استعمال المصفوفات، الدوال، break / continue

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;
    printf ("Enter The total number of registered students, N");
    scanf ("%d", &N);
    printf ("Enter The minimum attendance required, A");
    scanf ("%d", &A);
    printf ("Enter The absence threshold, S");
    scanf ("%d", &S);
    if (X < A) {
        printf ("The student is considered absent ");
    } else {
        printf ("The student is present");
    }
    for (i = 1, j = 1; i <= N; i++) {
        if (S >= i) {
            j++;
        }
    }
    if (j >= N) {
        printf ("Session valid");
    } else {
        printf ("Session cancelled");
    }
    return 0;
}
```

Copy 11

```
#include <stdio.h>
int main()
{
    int N, A, S, x;
    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("The student is considered absent");
    }

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

    int i, j, step;
    while (i <= N)
    {
        // ...
    }

    if (x>= S)
    {
        printf("Session Valid");
    }

    else
    {
        printf("Session cancelled");
    }

    return 0;
}
```

COPY NUMBER: 11

Analyse :

Algorithmique :

- Lectures correctes.
- Utilise `x` avant de le lire (dans le premier `if` hors boucle).
- Boucle `while (i <= N) vide (// ...)`. Code non terminé.
- Affichage final basé sur `x`.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initialisation	1 / 3	<code>x</code> non initialisé.
Condition boucle	0 / 4	Boucle vide.
Logique prés./abs.	1 / 4	Hors boucle et prématurée.
Compteurs	0 / 3	-
Affichages inter.	0 / 2	-
Affichage final	0 / 1	-

NOTE FINALE : 05 / 20

Feedback :

- **Points forts** : Début correct.
- **Points faibles** : Copie non finie. La logique principale est absente.
- **Appréciation globale** : Insuffisant.

Copy number : 12

تعليمات إلزامية : كتابة البرنامج كاملاً داخل main | استعمال حلقة واحدة فقط | يمنع استعمال المصفوفات، الدوال، break / continue

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, Absent, Present, Sessions;
    printf ("Enter the number of registered Students:");
    scanf ("%d", &N);
    printf ("Enter the minimum attendance registered:");
    scanf ("%d", &A);
    printf ("Enter the absence threshold:");
    scanf ("%d", &S);
    for (int i=0; i <= N || i <= S; i++) {
        printf ("Student number: %d", i);
        printf ("Number of attended Sessions:");
        scanf ("%d", &X);
        if (X < A) {
            printf ("Absent");
            Absent += i;
            printf ("Absent number: %d", Absent);
        }
        else {
            printf ("Present");
            Present += i;
            printf ("Present number: %d", Present);
        }
    }
    printf ("student number: %d", i);
    printf ("Present student : %d", Present);
    printf ("Absent student : %d", Absent);
    if (Session >= S) {
        printf ("Sessions cancelled");
    }
    else {
        printf ("Sessions Valid");
    }
}
```

Copy 12

```
#include <stdio.h>
int main()
{
    int N, A, S, X, Absent, Present, Session;
    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 == S; i++)
    {
        printf("Student number: %d", i);
        printf("Number of attended sessions");
        scanf("%d", &X);
        if (X < A)
        {
            printf("Absent");
            Absent += 1;
            printf("Absent number: %d", Absent);
        }
        else
        {
            printf("Present");
            Present += 1;
            printf("Present number: %d", Present);
        }
    }

    printf("student number: %d", i);
    printf("Present student: %d", Present);
    printf("Absent student: %d", Absent);
    if (Session>= S)
    {
        printf("Session cancelled");
    }
    else
    {
        printf("Session Valid");
    }
}
```

COPY NUMBER: 12

Analyse :

Algorithmique :

- Lectures correctes.

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

- i == S dans une condition de continuation est dangereux. Probablement voulu dire ... && Absent < S.
- i redéclaré dans la boucle (déjà utilisé pour le numéro étudiant, masquage possible si i était externe, mais ici i est le compteur).

- Comparaison if (X < A) correcte.

Incrémantation : Absent += 1, Present += 1 correcte.

- Attention : Absent et Present **non initialisés** ! Ils contiennent des valeurs aléatoires. Absent += 1 donnera n'importe quoi.

- Affichage final utilise une variable Session non initialisée (if (Session >= S)). Devrait être Absent.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initialisation	0 / 3	Variables d'accumulation non initialisées.
Condition boucle	2 / 4	Condition d'arrêt incorrecte (i == S au lieu de tester le compteur d'absents).
Logique prés./abs.	4 / 4	Correcte.
Compteurs	2 / 3	Logique d'incrémantation bonne, mais sur variables poubelles.
Affichages inter.	2 / 2	Présents.
Affichage final	0 / 1	Utilise une variable inconnue Session.

NOTE FINALE : 13 / 20

Feedback :

- Points forts** : Structure lisible, logique conditionnelle acquise.
- Points faibles** : **Initialisation des variables** ! C'est une erreur critique en C. Confusion sur les conditions d'arrêt.

- **Appréciation globale : Moyen.**
-

Copy number :

13

تعليمات إلزامية : كتابة البرنامج كاملاً داخل main | استعمال حلقة واحدة فقط | يمنع استعمال المصفوفات، الدوال، break / continue

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, $,
    Printf("Enter the total number: ");
    Scanf("%d", &total);
    Printf("Enter the minimum number of registered student: ");
    Scanf("%d", &minnum);
    Printf("Enter the absence threshold: ");
    Scanf("%d", absencethreshold);

    Printf("Enter the number of attended sessions X: ");
    while if (X < A) { while (X > A)
        X = absent;
        Printf("Absent");
    else
        X = Present;
    }
    i++;
    Printf("the student number: ");
    X = Present;
    Printf("else");
    X = Absent;
}

Printf("Enter the final status of exam: ");
return 0;
```

Copy 13

```
#include <stdio.h>
int main()
{
    int N, A, S;
    printf("Enter the total number: ");
    scanf("%d", &total);
    printf("Enter the minimum number of registered student: ");
    scanf("%d", &min_num);
    printf("Enter the absence threshold: ");
    scanf("%d", &absence_threshold);
    printf("Enter the number of attended sessions X: ");
    while (if (X <A))
    {
        X = absent;
    }

    else
    {
        X = Present;
    }

    printf("the student number: ");
    X = Present;
    else
    {
        X = Absent;
    }

    printf("Enter the final status of exam: ");
    return 0;
}
```

COPY NUMBER: 13

Analyse :

Algorithmique :

- Erreurs syntaxe `scanf: &total, &min_num...`. Les variables déclarées sont `N`, `A`, `S`. Noms incohérents.
- Boucle `while (if (x < A))`: Syntaxe invalide. Pas de `if` dans une condition `while` de cette manière.
- Affectations incorrectes : `x = absent`. On écrase la saisie ? `absent` et `Present` semblent être utilisés comme des variables mais non déclarées.
- Arguments du `scanf` manquants (pour `S`?).
- Code très confus.

Notation :

Critère	Points	Commentaire
Lecture <code>N, A, S</code>	1 / 3	Noms de variables incohérents avec la déclaration.
Initialisation	0 / 3	-
Condition boucle	0 / 4	Syntaxe invalide.
Logique prés./abs.	0 / 4	Incompréhensible.
Compteurs	0 / 3	-
Affichages inter.	0 / 2	-
Affichage final	0 / 1	-

NOTE FINALE : 01 / 20

Feedback :

- **Appréciation globale : Très Insuffisant.** Problèmes majeurs de syntaxe et de cohérence. Revoir les bases impérativement.

Copy number :

14

تعليمات إلزامية : كتابة البرنامج كاملاً داخل main | استعمال حلقـة واحدة فقط | يمنع استعمال المصفوفات، الدوال، break / continue

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,X;
    printf ("Enter the number of registered students = ");
    scanf ("%d", &N);
    printf ("Enter the number of attendances required = ");
    scanf ("%d", &A);
    printf ("Enter the absence threshold = ");
    scanf ("%d", &S);
    for (int i=0; i >= N || i < S; i++) {
        printf ("Student number for student: ");
        printf ("Number of student absences: ");
        scanf ("%d", &X);
        if (X < A) {
            printf ("Absent");
            printf ("Absent number = %d", i);
            printf ("A student is absent; %d% Absent");
        } else {
            printf ("Present");
            printf ("Present");
            printf ("Present, %d Present");
        }
        printf ("Student number: %d", i);
        printf ("Present student, %d Absent");
        if (S < i) {
            printf ("Complied");
        } else {
            printf ("Violated");
        }
    }
    return 0;
}
```

Copy 14

```
#include <stdio.h>
int main()
{
    int N, S, A, X;
    printf("Enter the number of registered students: ");
    scanf("%d", &N);
    printf("Enter the number of attendance required: ");
    scanf("%d", &A);
    printf("Enter the absence threshold: ");
    scanf("%d", &S);
    for (int i = 0; i <= N || i == S; i++)
    {
        printf("Student number of student: ");
        printf("Number of student hours: ");
        scanf("%d", &X);
        if (X < A)
        {
            printf("Absent");
            Absent += i;
            printf("Absent number: %d", Absent);
        }
        else
        {
            printf("Present");
            Present += i;
            printf("Present number: %d", Present);
        }
    }

    printf("Student number: %d", i);
    printf("Present student: %d", Present);
    if (Session == S)
    {
        printf("Cancelled");
    }
    else
    {
        printf("Valid");
    }
}

return 0;
}
```

COPY NUMBER: 14

Analyse :

Algorithmique :

- Lectures correctes.
- Boucle `for` avec condition `i <= N || i == S`. Arrêt sur `S` (seuil absences) incorrect car comparé à `i` (compteur tour).
- Saisie `scanf("%d", &X)`.
- Incrémentation : `Absent += i`. **Erreur** : Ajoute le numéro de l'étudiant (`i`) au lieu de 1 !
- `Absent` et `Present` non initialisés.
- Affichage final utilise `Session` (non déclaré).

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initialisation	0 / 3	Non faite.
Condition boucle	1 / 4	Condition d'arrêt sur seuil incorrecte.
Logique prés./abs.	4 / 4	Correcte.
Compteurs	0 / 3	Ajoute <code>i</code> au lieu de 1.
Affichages inter.	2 / 2	Présents.
Affichage final	0 / 1	Variable inexistante.

NOTE FINALE : 10 / 20

Feedback :

- **Points forts** : Code structuré qui ressemble à une solution.
- **Points faibles** : Incrémentation fausse (`+= i`), défaut d'initialisation, variables fantômes (`Session`).
- **Appréciation globale** : **Moyen -**

Copy number : 15

تعليمات إلزامية : كتابة البرنامج كاملاً داخل main | استعمال حلقة واحدة فقط | يمنع استعمال المصفوفات، الدوال، break / continue

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, P = 0, T = 0, i = 1;
    printf("Enter the number of registered student\n");
    scanf("%d", &N);
    printf("Enter the maximum attendance required\n");
    scanf("%d", &A);
    printf("Enter the absence threshold\n");
    scanf("%d", &S);
    while(i < N && T < S) {
        printf("Enter the number of attended session\n");
        scanf("%d", &X);
        if(X < A) {
            T = T + 1;
        } else {
            P = P + 1;
        }
        printf("the number of the students is %d", i);
        printf("the number of present student is %d", P);
        printf("the number of absent student is %d", T);
        i++;
    }
    printf("the total number of absent student is %d", T);
    printf("the total number of present student is %d", P);
    if(S >= T) {
        printf("The session valid");
    } else {
        printf("The session cancelled");
    }
}
```

```
printf("The total processed student is %d", i);
return 0;
```

Copy 15

```
#include <stdio.h>
int main()
{
    int N, A, S, X, P = 0, T = 0, i = 1;
    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);
    while (i <= N && T < S)
    {
        printf("Enter the number of attended sessions\n");
        scanf("%d", &X);
        if (X < A)
        {
            T = T + 1;
        }
        else
        {
            P = P + 1;
        }

        printf("the number of the students is %d", i);
        printf("the number of present student is %d", P);
        printf("the number of absents student is %d", T);
        i++;
    }

    printf("the total number of absent student is %d", T);
    printf("the total number of present student is %d", P);
    if (S >= T)
    {
        printf("The session valid");
    }
    else
    {
        printf("the session cancelled");
    }

    printf("the total processed student is %d", i);
    return 0;
}
```

COPY NUMBER: 15

Analyse :

Algorithmique :

- Initialisation correcte ($P=0$, $T=0$).
- Boucle while ($i \leq N \ \&\& \ T < S$) : **Excellent condition.** Gère les deux cas d'arrêt avec un ET logique.
- Saisie et tests corrects.
- Incrémantation correcte.
- Affichages complets.
- Décision finale correcte.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initialisation	3 / 3	Correct.
Condition boucle	4 / 4	Parfaite ($\&\&$).
Logique prés./abs.	4 / 4	Correct.
Compteurs	3 / 3	Correct.
Affichages inter.	2 / 2	Correct.
Affichage final	1 / 1	Correct.

NOTE FINALE : 20 / 20

Feedback :

- Points forts** : Code parfait. Respect total des contraintes et de la logique. Bravo.
- Appréciation globale : Très Bon.**

Copy number : 16

تعليمات إلزامية : كتابة البرنامج كاملاً داخل main | يمنع استعمال حلقة واحدة فقط | يمنع استعمال المصفوفات، الدوال، break / continue

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;
    printf("enter the number of attended between x: ");
    scanf("%d", &x);
    char absent, present;
    if (x < A) {
        printf("the student is considered absent");
    } else
        printf("the student is present");
    } int count, na, nb, i = 1;
    while (i <= N)
    printf("enter the number of present student and absent student");
    na = N - na;
    nb = N - nb;
    count++;
    printf("%d", na);
    printf("%d", nb);
    int N, S;
    printf("enter the simulation step");
    if (all students are present or the number of absent student reaches S){
        printf("stop");
    } else
        printf("Continue");
    int student number, present students, absent students;
    printf("enter student number and present students and absent students");
    scanf("%d %d %d", &student number, &present students, &absent students);
    printf("enter present students");
    printf("enter absent students");
    printf("enter the total present student");
    printf("enter the final status");
    if (S > S) {
        printf("Session valid");
    } else
        printf("Session cancelled");
}
```

Copy 16

```
#include <stdio.h>
int main()
{
    int N, A, S, x;
    printf("enter the number of attended sessions x: ");
    scanf("%d", &x);
    char absent, present;
    if (x < A)
    {
        printf("enter the student is considered absent: ");
    }

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

    int count, na, nb, i = 1;
    while (i <= N)
    {
        printf("enter the number of present student and absent student");
        na = N - na;
        nb = N - nb;
        count++;
        printf("%d", na);
        printf("%d", nb);
    }

    int N, S;
    printf("enter the simulation stop: ");
    if (all students are processed or the number of absent student reaches s)
    {
        printf("stop");
    }

    else
    {
        printf("continue");
    }

    int student number, present students, absent students;
    printf("enter student number and present students and absent students");
    scanf("%d %d %d", &student_number, &present_students, &absent_students);
    printf("enter present students");
    printf("enter absent students");
    printf("enter the total processed student");
    printf("enter the final status");
    if (x>= S)
    {
```

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

else
{
    printf("Session cancelled");
}

return 0;
}
```

COPY NUMBER: 16

Analyse :

Algorithmique :

- Déclaration variables string/char absent, présent inutiles ou mal utilisées.
- Condition `if (x < A)` avant la boucle et avant lecture cohérente.
- Boucle `while (i <= N)`. Corps de boucle : `na = N - na`. Calculs mathématiques étranges au lieu d'incrémentation simple.
- Saisie `scanf` pour tous les résultats à la fin ?? ("enter student number..."). L'étudiant redemande les résultats à l'utilisateur au lieu de les calculer.
- Pseudo-code dans les conditions (`if all students are processed...`).

Notation :

Critère	Points	Commentaire
Lecture N, A, S	2 / 3	Partiel.
Initialisation	0 / 3	-
Condition boucle	1 / 4	Boucle sur N uniquement.
Logique prés./abs.	1 / 4	Confuse.
Compteurs	0 / 3	Formules incorrectes.
Affichages inter.	0 / 2	-
Affichage final	0 / 1	Demande à l'utilisateur de saisir le résultat !

NOTE FINALE : 04 / 20

Feedback :

- **Points faibles** : Utilisation de pseudo-code. Ne calcule pas les résultats mais demande à l'utilisateur de les entrer à la fin.
- **Appréciation globale : Très Insuffisant.**

Copy number :

١٧

تعليمات إلزامية : كتابة البرنامج كاملاً داخل main | استعمال حلقة واحدة فقط | يمنع استعمال المصفوفات، الدوال، break / continue

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 ("N : ");
    scanf ("%d", &N);
    printf ("A : ");
    scanf ("%d", &A);
    printf ("S : ");
    scanf ("%d", &S);
    i = 1;
    while (i < N || i == S) {
        if (x < A) {
            printf ("the student is considered absent");
            printf (" Session cancelled");
        } else {
            printf ("the student is present");
            printf (" Session Valid");
        }
        i++;
    }
    return 0;
}
```

Copy 17

```
#include <stdio.h>
int main()
{
    int N, A, S;
    printf("N: ");
    scanf("%d", &N);
    printf("A: ");
    scanf("%d", &A);
    printf("S: ");
    scanf("%d", &S);
    int i = 1;
    while (i <N || i == S)
    {
        if (X <A)
        {
            printf("the student is considered absent");
            printf("session cancelled");
        }
        else
        {
            printf("the student is present");
            printf("session valid");
        }
    }
    return 0;
}
```

COPY NUMBER: 17

Analyse :

Algorithmique :

- Boucle `while (i < N || i == S)`. Condition d'arrêt incorrecte et mauvaise variable (`i` vs compteur d'absents).
- Utilisation de `X` sans lecture préalable dans la boucle (pas de `scanf`).
- `return 0` dans la boucle.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initialisation	1 / 3	Variables par défaut.
Condition boucle	1 / 4	Incorrecte.
Logique prés./abs.	1 / 4	Pas de lecture de X.
Compteurs	0 / 3	-
Affichages inter.	0 / 2	-
Affichage final	0 / 1	-

NOTE FINALE : 06 / 20

Feedback :

- **Appréciation globale : Insuffisant.** Programme qui ne fait rien (pas de lecture de données dans la boucle).

Copy number :

18

تعليمات إلزامية : كتابة البرنامج كاملاً داخل main استعمال حلقة واحدة فقط | يمنع استعمال المصفوفات، الدوال، break / continue

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;
    int i = 1;
    printf("entre N: \n");
    scanf("%d", &N);
    printf("entre A and S: \n");
    scanf("%d %d", &A, &S);
    while (i <= N) {
        int x;
        printf("entre x: \n");
        scanf("%d", &x);
        if (x < A) {
            printf("Student is absent");
        } else if (i == S) {
            return 0;
        } else {
            printf("Student is present");
        }
        i++;
    }
    printf("present students %d: \n", present students);
    printf("absent students %d: \n", absent students);
    if (absent students > A) {
        printf("ذئابون");
    } else {
        printf("ذئاب");
    }
    return 0;
}
```

Copy 18

```
#include <stdio.h>
int main()
{
    int N, A, S;
    int i = 1;
    printf("entre N: \n");
    scanf("%d", &N);
    printf("entre A and S: \n");
    scanf("%d %d", &A, &S);
    while (i <= N)
    {
        int x;
        printf("entre x: \n");
        scanf("%d", &x);
        if (x < A)
        {
            printf("Student is absent");
            if (i == S)
            {
                return 0;
            }
        }
        else
        {
            printf("student is present");
        }

        i++;
        printf("present students %d: \n", present_students);
        printf("absent students %d: \n", absent_students);
        if (absent_students > A)
        {
            printf("Exam cancelled");
        }
        else
        {
            printf("Exam valid");
        }
    }
    return 0;
}
```

COPY NUMBER: 18

Analyse :

Algorithmique :

- Lectures correctes.
- Boucle while ($i \leq N$).
- Condition d'arrêt if ($i == S$) : Compare indice boucle au seuil d'absence (Faux). De plus, fait un return 0 brutal. (Interdiction de break/continue contournée par return ou if structurant? return quitte le prog, donc plus d'affichage final).
- Calculs : Compare absent_students > A pour annulation ? C'est > S. Utilise des variables non déclarées/init (absent_students).

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initialisation	1 / 3	Variables manquantes.
Condition boucle	2 / 4	Boucle sur N ok, arrêt seuil mal géré.
Logique prés./abs.	3 / 4	Correcte.
Compteurs	0 / 3	Variables non déclarées.
Affichages inter.	1 / 2	Présent.
Affichage final	0 / 1	Incorrect.

NOTE FINALE : 10 / 20

Feedback :

- **Appréciation globale : Moyen -**. Logique présente mais implémentation défaillante (variables, conditions).

Copy number :

19

تعليمات إلزامية : كتابة البرنامج كاملاً داخل main | يمنع استعمال حلقة واحدة فقط | يمنع استعمال المصفوفات، الدوال، break / continue

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

    int i, X[10], p = 0, a = 0; // p: present students, a = absent students
    for (i = 1; i <= N || a > S; i++) {
        printf ("%d = ", i);
        scanf ("%d", &X[i]);
        if (X[i] < A) {
            printf ("%d absent", i);
            a++;
        } else {
            printf ("%d present", i);
            p++;
        }
        printf ("a = %d", a);
        printf ("p = %d, p", p);
    }
    if (a < S) {
        printf ("Session Invalid");
    } else {
        printf ("Session Cancelled");
    }
    return 0;
}
```

Copy 19

```
#include <stdio.h>
int main()
{
    int N, A, S;
    printf("N = ");
    scanf("%d", &N);
    printf("A = ");
    scanf("%d", &A);
    printf("S = ");
    scanf("%d", &S);
    int i, X, P = 0, a = 0;
    // p: present students, a: absent for (i = 1; i <= N || a > S; i++)
    {
        printf("%d", i);
        scanf("%d", &X);
        if (X < A)
        {
            printf("%d absent", i);
            a++;
        }
        else
        {
            printf("%d present", i);
            P++;
        }
    }

    printf("a = %d", a);
    printf("p = %d", P);
}

if (a < S)
{
    printf("Session Valid");
}

else
{
    printf("session cancelled");
}

return 0;
}
```

COPY NUMBER: 19

Analyse :

Algorithmique :

- Initialisation $P=0$, $a=0$.
- Boucle `for (i = 1; i <= N || a > S; i++)` : Condition `||` (OU) signifie que la boucle continue SI on dépasse le seuil !! C'est l'inverse (`&&` $a < S$).
- Lectures et tests corrects.
- Incrémantation correcte.
- Affichage final correct.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initialisation	3 / 3	Correct.
Condition boucle	2 / 4	Erreur de logique booléenne (<code> </code> prolonge la boucle au lieu d'arrêter).
Logique prés./abs.	4 / 4	Correct.
Compteurs	3 / 3	Correct.
Affichages inter.	2 / 2	Correct.
Affichage final	1 / 1	Correct.

NOTE FINALE : 18 / 20

Feedback :

- Points forts** : Très bon code.
- Points faibles** : Petite erreur de logique sur la condition d'arrêt (`||` vs `&&`).
- Appréciation globale** : Très Bon.

Copy number :

20

تعليمات إلزامية : كتابة البرنامج كاملاً داخل main | استعمال حلقة واحدة فقط | يمنع استعمال المصفوفات، الدوال، break / continue

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, absent = 0, present = 0, N';
    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);
    for (int i = 1; i <= N; i++)
    {
        printf("enter the number of attended\n");
        N[i] = i;
        scanf("%d", &X);
        if (A > X)
        {
            printf("The student is absent");
            absent++;
            printf("The Number of student absent\n");
            N[i] = absent;
        }
        else
        {
            printf("The student is present");
            present++;
            printf("The number of students\n");
            present N[i] = present;
        }
        if (absent == S)
        {
            printf("The simulation stop");
        }
    }
    printf("The total processed students\n");
    i % d, i);
    printf("The total present students\n");
    is: % d, present);
    printf("The total absent students\n");
    is: % d, absent);
    N' = N - A;
    if (N' > S)
    {
        printf("Session Cancelled");
    }
    else
    {
        printf("Session Valid");
    }
    return 0;
}
```

Copy 20

```
#include <stdio.h>
int main()
{
    int N, A, S, X, absent = 0, present = 0;
    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);
    for (int i = 1; i <= N; i++)
    {
        printf("enter the number of attended sessions: ");
        scanf("%d", &X);
        if (A > X)
        {
            printf("The student is absent");
            absent++;
            printf("The number of student absent now is: %d", absent);
        }
        else
        {
            printf("The student is present");
            present++;
            printf("The number of students present now is: %d", present);
        }

        if (absent == S)
        {
            printf("The simulation stop");
        }

        printf("The total processed students is %d", i);
        printf("The total present students is: %d", present);
        printf("The total absent students is: %d", absent);
        if (absent >= S)
        {
            printf("Session Cancelled");
        }

        else
        {
            printf("Session Valid");
        }
    }

    return 0;
}
```


COPY NUMBER: 20

Analyse :

Algorithmique :

- Initialisation correcte.
- Boucle `for` OK. Pas d'arrêt prématuré explicite dans la condition du `for` (sauf `i<=N`), mais un `if (absent == S)` avec un message. Mais pas de `break` (interdit) ni modif de `i`. Donc la boucle continue.
- `return 0` dans la boucle après affichages finaux ? Le `return` est à la fin du bloc `if/else final`, qui est DANS la boucle ? Non, indentation suggère hors boucle.
- Logique propre.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initialisation	3 / 3	Correct.
Condition boucle	2 / 4	Ne gère pas l'arrêt effectif sur seuil (affiche juste le message).
Logique prés./abs.	4 / 4	Correct.
Compteurs	3 / 3	Correct.
Affichages inter.	2 / 2	Correct.
Affichage final	1 / 1	Correct.

NOTE FINALE : 18 / 20

Feedback :

- **Appréciation globale : Très Bon.**

Copy number :

21

تعليمات إلزامية : كتابة البرنامج كاملاً داخل main | استعمال حلقة واحدة فقط | يمنع استعمال المصفوفات، الدوال، break / continue

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 i; int N; int S; int A; int X
    for (int x, i=0, i <= X, i++) {
        if (X < A)
            printf ("the student is considered absent");
        else
            printf ("the student is present");
        if (all N student are prosseced or the number of absent student
            research S)
            printf (simulation stop);
        else
            printf (continous the simulation);
        if (S >= A)
            printf (session valid).
        else
            printf (session cancelled); }
    return 0; }
```

Copy 21

```
#include <stdio.h>
int main()
{
    int i, N, S, A, X;
    for (int i = 0, i <= X, i++)
    {
        if (X < A)
        {
            printf("the student is considered absent");
        }

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

        if (all N student are processed or the number of absent student research S)
        {
            printf("simulation stop");
        }

        else
        {
            printf("continue the simulation");
        }

        if (S>= A)
        {
            printf("session valid");
        }

        else
        {
            printf("session cancelled");
        }
    }

    return 0;
}
```

COPY NUMBER: 21

Analyse :

Algorithmique :

- Boucle `for (int i = 0, i <= x, i++)` : Condition sur x (non initialisé) !
- Pseudo-code dans la condition `if (all N student are processed...)`.
- Code très incomplet.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	0 / 3	Absente.
Initialisation	0 / 3	-
Condition boucle	0 / 4	Incorrecte.
Logique prés./abs.	1 / 4	Sommaire.
Compteurs	0 / 3	-
Affichages inter.	0 / 2	-
Affichage final	0 / 1	-

NOTE FINALE : 01 / 20

Feedback :

- Appréciation globale : Très Insuffisant.

Copy number : 22

تعليمات إلزامية : كتابة البرنامج كاملاً داخل main | استعمال حلقة واحدة فقط | يمنع استعمال المصفوقات، الدوال، break / continue

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 ("Enter total number of registered students");

scanf ("%d", &N);

printf ("minimum attendance required");

scanf ("%d") &A);

printf ("absence threshold");

scanf ("%d") &S);

for (i = 1; i >= N; i++) {

{ if (x < A)

printf ("The student is absent");

else

printf ("The student is present");}

if (A >= N)

printf ("session valid");

else

printf ("session cancelled");

}

return 0;

}

Copy 22

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

    if (A>= N)
    {
        printf("session valid");
    }
    else
    {
        printf("session cancelled");
    }
}
```

COPY NUMBER: 22

Analyse :

Algorithmique :

- Lectures correctes.
- Utilise `x` sans le lire (`scanf` absent).
- Pas de compteurs.
- Condition finale `if (A >= N)` incohérente.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	Correct.
Initialisation	1 / 3	Incomplète.
Condition boucle	1 / 4	Basique.
Logique prés./abs.	1 / 4	Pas de saisie de données.
Compteurs	0 / 3	-
Affichages inter.	0 / 2	-
Affichage final	0 / 1	Incorrect.

NOTE FINALE : 06 / 20

Feedback :

- **Appréciation globale : Insuffisant.** Pas fonctionnel.

Copy number : 23

تعليمات إلزامية : كتابة البرنامج كاملاً داخل main | استعمال حلقة واحدة فقط | يمنع استعمال المصفوفات، الدوال، break / continue

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 i;
    int N, A, S;
    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; i++) {
        printf("Enter Student number");
        printf("Enter X");
        scanf("%d", &X);
        if (X < A) {
            printf("The student absent");
        } else {
            printf("The student present");
        }
        absent Students = N - present Students;
        printf("%d", absent Student \n);
        present Students = N - absent Students;
        printf("%d", present Students);
        printf("%d", present Students);
        printf("%d", present Students);
        if (absent Students == S) {
            printf("Session cancelled");
        }
    }
}
```

Copy 23

```
#include <stdio.h>
int main()
{
    int i, x, N, A, S;
    printf("Enter N");
    scanf("%d", &N);
    printf("Enter A");
    scanf("%d", &A);
    printf("Enter S");
    scanf("%d", &S);
    for (i = 1; i <= N || i == S; i++)
    {
        printf("enter student number");
        printf("enter X");
        scanf("%d", &x);
        if (x < A)
        {
            printf("The Student absent");
        }
        else
        {
            printf("The Student present");
        }

        absent students = N - present students;
        printf("absent student: %d", absent_students);
        present students = N - absent students;
        printf("present students: %d", present_students);
        if (absent students == 5)
        {
            printf("Session cancelled");
        }
        else
        {
            printf("session valid");
        }
    }
}
```

COPY NUMBER: 23

Analyse :

Algorithmique :

- Boucle `for (i = 1; i <= N || i == S; i++)` : Condition d'arrêt sur `i == S` (indice vs seuil).
- Calculs absent students = N - present students mais present students non initialisé.
- Logique circulaire.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Initialisation	0 / 3	Manquante.
Condition boucle	2 / 4	Erreur logique indice/seuil.
Logique prés./abs.	2 / 4	OK pour la condition.
Compteurs	0 / 3	Calculs faux.
Affichages inter.	1 / 2	OK.
Affichage final	0 / 1	OK.

NOTE FINALE : 08 / 20

Feedback :

- Appréciation globale : Fragile.

Copy number : ٢٤

تعليمات إلزامية : كتابة البرنامج كاملاً داخل main | استعمال حلقة واحدة فقط | يمنع استعمال المصفوفات، الدوال، break / continue

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, N, S;
    printf ("read the number of attended sessions x");
    scanf ("%d", &x);
    "rules for each student"
    if (x < 0)
        printf ("The student is considered absent");
        scanf ("%d\n", absent student);
    else
        printf ("the student is considered present");
        scanf ("%d\n", present student);
    }
    printf ("%d\n" x);
```

stop conditions

```
for (i = N) i = 5 {
    ("i = N; i < N; i++");
    ("j = S; j <= N; j++");
    printf ("Session valid");
    printf ("Session canceled");
```

{ return 0;

Copy 24

```
#include <stdio.h>
int main()
{
    int A, N, S, x;
    printf("read the number of attended sessions x");
    scanf("%d", &x);
    // rules for each student if (x <0)
    {
        printf("The student is considered absent");
        scanf("%d", absents_student);
    }

    else
    {
        printf("the student is considered present");
        scanf("%d", present_student);
    }

    printf("%d", x);
    // stop conditions for (i = N)
    {
        // ...
    }

    printf("Sersion valid");
    printf("Sersion canceled");
    return 0;
}
```

COPY NUMBER: 24

Analyse :

Algorithmique :

- Code fragmentaire.
- `if (x < 0)` : Condition étrange (devrait être A).
- `scanf("%d", absents_student)` : Utilise scanf en écriture ??
- Boucle `for (i = N)` syntaxiquement fausse.

NOTE FINALE : 02 / 20

Feedback :

- Appréciation globale : Très Insuffisant.