

Copy number : 1

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, J, T = 0;
    i = 0; G = 0; Z;
    scanf("%d", &N);
    scanf("%d", &A);
    scanf("%d", &S);
    while (T != S) {
        while (i != N) {
            int H = 0;
            scanf("%d", &Z);
            for (J = 0; J < Z; J++) {
                int f;
                scanf("%d", &f);
                if (f == 1)
                    H++;
            }
        }
    }
}
```

```
if (x < A)
    T = T + 1;
else
    G = G + 1; i++;
printf("%d", G);
printf("%d", T);
printf("المتوسط صاج");
}
printf("المتوسط صاج");
return 0;
}
```

Copy 1

```
#include <stdio.h>
int main()
{
    int N, A, S, J, T = 0;
    int i = 0, G = 0, Z;
    scanf("%d", &N);
    scanf("%d", &A);
    scanf("%d", &S);
    while (T != S)
    {
        while (i <= N)
        {
            int H = 0;
            printf("Enter x");
            scanf("%d", &Z);
            for (J = 0; J <Z; J++)
            {
                int f;
                scanf("%d", &f);
                if (f == 1)
                {
                    H++;
                }
            }

            if (x <A)
            {
                T = T + 1;
            }

            else
            {
                G = G + 1;
                i++;
            }

            printf("%d %d", G, T);
        }

        printf("Session valid");
    }

    printf("Session cancelled");
    return 0;
}
```

Analyse :**Algorithmique :**

- Boucles imbriquées `while (T != S) { while (i <= N) { ... } ... }`. Structure très lourde et incorrecte (boucle infinie si `T != S`).
- Boucle interne `for (J=0; J<Z; J++)` : Itère sur Z (valeur lue) ? Semble simuler quelque chose d'autre que la présence (peut-être un tableau de présences par séance ?).
- Logique complexe et confuse.

Notation :

Critère	Points	Commentaire
Lecture N, A, S	3 / 3	OK.
Initialisation	1 / 3	Beaucoup de variables inutiles.
Condition boucle	1 / 4	Boucles imbriquées inutiles et dangereuses.
Logique prés./abs.	2 / 4	Conditions présentes mais noyées dans une mauvaise structure.
Compteurs	2 / 3	OK.
Affichages inter.	0 / 2	Confus.
Affichage final	1 / 1	OK.

NOTE FINALE : 10 / 20**Feedback :**

- **Appréciation globale : Moyen -.** Pourquoi tant de boucles ? Restez simple.
-

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

```
int main ( ) {
```

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

```
printf("Enter The Number of The Students = ");
```

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

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

```
printf("Enter The number of attended sessions = ");
```

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

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

```
printf("The student is absent\n");
```

```
else
```

```
printf("The student is present\n");
```

```
if (N == i && N < S)
```

```
printf("Stop Simulation");
```

```
else printf("The total Students");
```

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

```
printf("The present Students");
```

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

```
printf("The absent Students");
```

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

```
if (N > S)
```

```
printf("The session is valid");
```

```
else
```

```
printf("The session is cancelled");
```

```
return 0;
```

Copy 2

```
#include <stdio.h>
int main()
{
    int N, A, S;
    printf("Enter The Number of The Students: ");
    scanf("%d", &N);
    for (int i = 1; i <= N; i++)
    {
        printf("Enter The number of attended sessions: ");
        scanf("%d", &x);
        if (X < A)
        {
            printf("The Student is absent\n");
        }

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

        if (N == i && N < S)
        {
            printf("Stop Simulation");
            printf("The total Students");
            scanf("%d", &N);
            printf("The present Students");
            scanf("%d", &i);
            printf("The absent Students");
            scanf("%d", &S);
        }

    }

    if (N > S)
    {
        printf("The session is valid");
    }

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

    return 0;
}
```

Analyse :

Algorithmique :

- Boucle for simple.
- Logique interne correcte.
- Condition d'arrêt `if (N == i && N < S)` ? Test d'arrêt incorrect (`i` est l'indice, il vaudra `N` à la fin).
- Code de 'Stop Simulation' demande à l'utilisateur de resaisir `N`, `i`, `S` ? Contresens total.

NOTE FINALE : 12 / 20

Feedback :

- **Appréciation globale : Moyen.** Bonne structure mais fin incohérente (ressaisie des données).
-

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

```

#include <stdio.h>    int main() {
    int N, A, S, P, C, X, M = 0;
    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);
    C = N;
    P = 1;
    for(i = 0; i < N || i != S; i++) {
        printf("student number: %d", P);
        P++;
        printf("Enter the number of attended session");
        scanf("%d", &X);
        if(X < A)
            C = C - 1;
        M = N - C;
        printf("Present students: %d", C);
        printf("absent students: %d", M);
    }
    printf("Present students: %d", C);
    printf("absent student: %d", M);
    if(M < S)
        printf("Session valid");
    else
        printf("Session cancelled");
    return 0;
}

```

Copy 3

```
#include <stdio.h>
int main()
{
    int N, A, S, P, C, X, M = 0;
    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);
    C = N;
    P = 1;
    for (i = 0; i <N || i != S; i++)
    {
        printf("student number: %d", P);
        P++;
        printf("Enter the number of attended session");
        scanf("%d", &X);
        if (X <A)
        {
            C = C - 1;
            M = N - C;
            printf("Present students: %d", C);
            printf("absent students: %d", M);
        }

        else
        {
            printf("Present students: %d", C);
            printf("absent students: %d", M);
        }

    }

    if (M <S)
    {
        printf("Session valide");
    }

    else
    {
        printf("Session cancelled");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Boucle for (. . . | | i != S). Condition arrêt fausse (| | continue si i != S, donc tant que i != S la boucle tourne même si i > N).
- Compteurs : Initialise C = N. Décrémente si absent (C = C - 1). M = N - C (absents).
- Logique cohérente (compte à rebours des présents).

NOTE FINALE : 14 / 20

Feedback :

- **Appréciation globale : Moyen / Bon.** Approche originale (compte à rebours).
-

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, B, n;

    printf("Entre the total number of registered students N:");
    scanf("%d", &N);
    printf("Entre the minimum attendance required A:");
    scanf("%d", &A);
    printf("Entre the absence threshold S:");
    scanf("%d", &S);

    for (int i = 1; i <= N; i++) {
        printf("read the number of attended sessions n:");
        scanf("%d", &n);
        if (n < A) {
            printf("the student is considered is absent");
        }
        else {
            printf("the student is considered is present");
        }
        B = B + 1;
        n = N - B;
    }
}

```

```

printf("the number of present is %d", B);
printf("the number of absence is %d", n);
}
if (B > A || n < S) {
    printf("session valide");
}
else {
    printf("session cancelled");
}
return 0;
}

```

Copy 4

```
int main()
{
    int N, A, S, B, n;
    printf("Entre the total number of registred students N: ");
    scanf("%d", &N);
    printf("Entre the minimum attendance required A: ");
    scanf("%d", &A);
    printf("Entre the absence threshold S: ");
    scanf("%d", &S);
    for (int i = 1; i <= N; i++)
    {
        printf("read the number of attended sessions x: ");
        scanf("%d", &n);
        if (n < A)
        {
            printf("the student is considered is absent");
        }

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

    }

    B = B + 1;
    n = N - B;
    printf("the number of present is %d", B);
    printf("the number of absente is %d", n);
    if (B > A || n < S)
    {
        printf("session Valide");
    }

    else
    {
        printf("session cancelled");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- `int B` non initialisé.
- Boucle `for`.
- Calculs finaux : $B = B + 1$ (?? Une seule fois hors boucle). $n = N - B$.
- Les compteurs ne sont pas mis à jour **DANS** la boucle. Donc `B` ne compte rien (valeur poubelle + 1).

NOTE FINALE : 06 / 20

Feedback :

- **Appréciation globale : Insuffisant.** Les compteurs doivent être dans la boucle.
-

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, SumPresent, Sumabsent=0, X;
    printf("مراقبة حضور الطلبة");
    do {
        scanf("%d", &X);
        if (X < A) {
            printf("عدد طالب الغائب");
            Sumabsent = Sumabsent + 1;
        } else {
            if (X > A) {
                printf("عدد الطالب الحاضر");
                SumPresent = SumPresent + 1;
            } else {
                i = i + 1;
            }
        }
        while (i <= N || Sumabsent >= S) {
            if (Sumabsent < S) {
                printf("الإمتحان صالح");
            } else {
                if (Sumabsent > S) {
                    printf("الإمتحان غير صالح");
                } else {
                    printf("الإمتحان غير صالح");
                }
            }
        }
        return 0;
    }
}

```

Copy 5

```
#include <stdio.h>
int main()
{
    int N, A, SumPresent = 0, Sumabsent = 0, X;
    printf("Entrer A");
    scanf("%d", &X);
    if (X < A)
    {
        printf("nombre d'absents");
        Sum_absent = Sum_absent + 1;
    }

    else
    {
        if (X > A)
        {
            printf("nombre de presents");
            Sum_present = Sum_present + 1;
        }
    }

    int i = i + 1;
    while (i <= N || Sum_absent >= S)
    {
        if (Sum_absent < S)
        {
            printf("L'examen est valide");
        }

        else
        {
            if (Sum_absent > S)
            {
                printf("L'examen est non valide");
            }
        }
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Lit X (une seule fois) **avant** la boucle (dans scanf("%d", &X) pour A ??). Confusion totale.
- Structure if/else avant la boucle.
- Boucle while infinie (i incrémenté une fois avant).

NOTE FINALE : 04 / 20

Feedback :

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

```

#include <stdio.h>

int main () {
    Var : N , i , A , S, x; integer ;
    Sum Present = 0 ;
    scanf ("%d", &A, &S); i = 0;
    Do {
        scanf ("%d", &N);
        scanf ("%d", &x);
        if (x < A) {
            printf ("الطالب غائب", i);
            Sum absent = Sum absent + 1; printf ("عدد الغائبين", Sum absent);
            Sum Present = Sum Present; printf ("عدد الحاضرين", Sum Present);
        }
        else
            printf ("الطالب حاضراً", i);
            Sum Present = Sum present + 1; printf ("عدد الحاضرين", Sum Present);
            Sum absent = Sum absent; printf ("عدد الغائبين", Sum absent);
    } while (i <= N || Sum absent >= S) {
        if (Sum absent >= S) {
            printf ("الامتحان غير صالح");
        }
        else
            printf ("الامتحان غير صالح");
    }
    return 0;
}

```

Copy 6

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

    else
    {
        printf("Etudiant present");
        SumPresent = SumPresent + 1;
        printf("Sum present: %d", SumPresent);
        printf("Sum absent: %d", Sumabsent);
    }

    while (i <= N || Sumabsent >= S)
    {
        if (Sumabsent >= S)
        {
            printf("L'examen non valid");
        }

        else
        {
            printf("L'examen est valid");
        }

        return 0;
    }
}
```

Analyse :

Algorithmique :

- Même erreur : Lit X une seule fois avant la boucle.
- Traite le cas présent/absent une fois.
- Boucle `while` qui répète le résultat final ?
- `return 0` dans la boucle.

NOTE FINALE : 05 / 20

Feedback :

- **Appréciation globale : Insuffisant.** Ne traite qu'un seul étudiant.
-

```

#include <studio.h>
int main() {
    int N, A, S, X, K=0, F=0, n;
    While (N != 0 & K != S) {
        printf("Entrez x");
        scanf("%d", &X);
        if (X < A) {
            printf("the student is considered absent");
            F = F + 1;
        }
        else {
            printf("the student is present");
            K = K + 1;
        }
    }
    printf("%d; present students; K);
    printf("%d; absent student; S);
    printf("%d = %d + %d", n = K + S);
    if (K = S) {
        printf("session cancelled");
    }
    else {
        printf("session Valid");
    }
    } return 0;

```

Copy 7

```
#include <stdio.h>
int main()
{
    int N, A, S, X, K = 0, J = 0, n;
    while (N != 0 && K != S)
    {
        printf("Entrer x");
        scanf("%d", &x);
        if (x < A)
        {
            printf("the student is considered absent");
            J = J + 1;
        }

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

    }

    printf("%d present students", K);
    printf("%d absent student", J);
    printf("%d = %d + %d", n = K + S);
    if (K == S)
    {
        printf("session cancelled");
    }

    else
    {
        printf("session valid");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Boucle `while (N != 0 && K != S)`. N est constant, donc `N != 0` toujours vrai. Arrêt sur `K (présents) == S (seuil absents)` ?? Confusion variables.
- Pas de compteur de boucle (boucle infinie, sauf si K atteint S).
- Logique interne correcte.

NOTE FINALE : 09 / 20

Feedback :

- **Appréciation globale : Fragile.** Boucle potentiellement infinie. Confusion sur la condition d'arrêt.
-

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

```
#include <stdio.h>
#include <stdlib.h>
int main ()
{
    int N, A, S, F, E, Y, O, X;
    int i;
    printf("Enter the total number of
    registered students: \n");
    scanf("%d", &N);
    printf("Enter the minimum attendance
    required: \n");
    scanf("%d", &A);
    printf("Enter the absence threshold");
    scanf("%d", &S);
    for (i = 1; i <= N; i++)
    {
        printf("Enter the number of attended
        sessions of the student %d : \n");
        scanf("%d", &X);
        if (X < A) // عدد الحضور (X) < عدد الحصة المطلوبة (A)
        {
            printf("The student %d is
            absent: \n", i);
            Y = Y + 1;
            printf("The number of absent student
            is: %d \n", Y);
            printf("The number of present student
            is: %d \n", E);
        }
    }
}
```

```
else
{
    printf("The student %d is present: \n", i);
    E = E + 1;
    printf("The number of absent students is: %d \n", Y);
    printf("The number of present students is: %d \n", E);
}
}
if (Y == S) // لوقت الحصة
{
    i = N;
}
O = Y + E; // عدد الطلبة الذين تم معالجتهم
printf("Total processed students
are: %d \n", O);
printf("The total number of absent
students are: %d \n", Y);
printf("The total number of present
students are: %d \n", E);
if (Y > S)
{
    printf("The session is valid");
}
else
{
    printf("The session is cancelled");
}
return 0;
}
```

Copy 8

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

        else
        {
            printf("The student %d is present: \n", i);
            E = E + 1;
            printf("the number of absent student is: %d \n", Y);
            printf("the number of present students is: %d \n", E);
        }

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

    O = Y + E;
    printf("Total processed students are: %d \n", O);
    printf("the total number of absent students are: %d \n", Y);
    printf("the total number of present students are: %d \n", E);
    if (Y > S)
    {
        printf("the session is valid");
    }

    else
```

```
{  
    printf("the session is cancelled");  
}  
  
return 0;  
}
```

Analyse :

Algorithmique :

- Boucle for OK.
- Logique interne OK.
- Arrêt : `if (Y == S) i = N.` (Force la fin de boucle). Correct.
- Variables non initialisées Y, E (accumulateurs). Grave en C.

NOTE FINALE : 13 / 20

Feedback :

- **Appréciation globale : Moyen.** Attention à l'initialisation.
-

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, C, i, T;
    printf("Enter the total number of registered students");
    scanf("%d", &N);
    printf("Enter the minimum attendance required");
    scanf("%d", &A);
    printf("Enter absence threshold");
    scanf("%d", &S);
    T = N;
    for (N = 0; N <= T) {
        int x;
        printf("Enter the number of attendance session of the student");
        scanf("%d", &x);
        if (x < A)
            i = 0;
        printf("the student is absent");
        i++;
        printf("the number of absent student is: %d\n", i);
        else
            C = 0;
        printf("the student is present");
        C++;
        printf("the number of present student is: %d\n", C);
    }
    if (i >= S)
        printf("session cancelled");
    else
        printf("session valid");
    Return 0;
}

```

Copy 9

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

        else
        {
            C == 0;
            printf("the student is present");
            C++;
            printf("the number of present student is: %d \n", C);
        }

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

        else
        {
            printf("session Valid");
        }

        return 0;
    }
}
```

Analyse :

Algorithmique :

- Boucle for (N = 0; N >= T). N (nombre étudiants) écrasé par 0. T = N initial.
- Condition N >= T (0 >= N) : Faux dès le début (sauf si N=0). La boucle ne s'exécute pas.
- Logique interne : i == 0 (comparaison inutile), i++ (sur variable non init).

NOTE FINALE : 03 / 20

Feedback :

- **Appréciation globale : Très Insuffisant.** Boucle ne démarre pas.
-

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 n");
    scanf("%d" & n);
    for (i=1, i <= n; i++) {
        printf("enter X");
        scanf("%d" & X);
        if (X < A) then {
            C = C + A
            printf("The student present");
        } else if (X > A) {
            C = C + 1;
            printf("The student absent");
        }
        if (S == S or N all processed) then {
            printf("simulation stop");
        }
    }
    if (A > S) then {
        printf("The session valid");
    } else { printf("session cancelled"); }
    return 0;
}

```

Copy 10

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

        else if (X > A)
        {
            C = C + 1;
            printf("The student absent");
        }

        if (S == S || N all processed)
        {
            printf("simulation stops");
        }

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

        else
        {
            printf("session cancelled");
        }

    }

    return 0;
}
```

Analyse :

Algorithmique :

- Incrémentation `C = C + A` (si absent ?). Bizarre.
- Condition arrêt `if (S == S || ...)` : Toujours vrai.
- Pseudo-code.

NOTE FINALE : 05 / 20

Feedback :

- Appréciation globale : Insuffisant.
-

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

```

#include <stdio.h>
int main() {
    int N, A, S;
    printf("Enter total number of registered students N");
    scanf("%d", &N);
    printf("Enter minimum attendance required A");
    scanf("%d", &A);
    printf("Enter absence threshold S");
    scanf("%d", &S);
    for (i = 1; i <= N; i++) {
        scanf("%d", &n);
        if (n < A) {
            m++;
            printf("absent", "%d/m", m);
        } else {
            n++;
            printf("Present", "%d/n", n);
            if (m < S) {
                printf("المسحوق!");
            }
        }
    }
    printf("المسحوق!");
    return 0;
}

```

Copy 11

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

        else
        {
            n++;
            printf("Present", n);
        }

        if (m < S)
        {
            printf("Session valid");
        }

    }

    printf("Session cancelled");
    // end program return 0;
}
```

Analyse :

Algorithmique :

- Lecture x sans &. (scanf ("%d", x) -> Crash).
- Variables m, n non initialisées.
- Structure correcte.

NOTE FINALE : 08 / 20

Feedback :

- **Appréciation globale : Fragile.** Erreur pointeur scanf.
-

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

include <stdio.h>

int main () {

int N, A, S, X;

int n, p, a;

int i=1, counter 1=0, counter 2=0;

printf ("Enter N: \n");

scanf ("%d", &N);

while (i <= N) {

i==n

printf ("student number n: \n");

printf ("Enter X, A");

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

if (X < A) {

printf ("session cancelled");

counter 2 += i;

i++; counter 2 == a;

printf ("absent students: \na");

else X > A {

printf ("session valid");

counter 1 += i;

i++; counter 1 == P;

printf ("present students: \np");

}

return 0; }

Copy 12

```
#include <stdio.h>
int main()
{
    int N, A, S, X;
    int n, p, a;
    int i = 1, counter1 = 0, counter2 = 0;
    printf("Enter N: \n");
    scanf("%d", &N);
    while (i <= N || a == S)
    {
        i == n;
        printf("student number n: %d", n);
        printf("Enter X, A");
        scanf("%d %d", &X, &A);
        if (X < A)
        {
            printf("session cancelled");
            counter2 += i;
            i++;
            counter2 == a;
            printf("absent students: %d", a);
        }

        else if (X > A)
        {
            printf("session valid");
            counter1 += i;
            i++;
            counter1 == p;
            printf("present students: %d", p);
        }

    }

    return 0;
}
```

Analyse :

Algorithmique :

- `while (i <= N || a == S). (|| continue si a == S).`
- Comparaisons pseudo-code `counter1 == p.` (Comparaison sans effet).
- Incrémentation `counter2 += i.` Ajoute l'indice étudiant ??
- Confus.

NOTE FINALE : 07 / 20

Feedback :

- **Appréciation globale : Insuffisant.**
-

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

#include <stdio.h>

int main () {

int N, A, S, C=0, B=0, i, P, 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);

while (i ≤ N && i ≤ S) {

. Switch :

case (i): {

printf("Enter the number of attended sessions of student: ", i);

scanf("%d", &X);

IF (X < A) {

| C = C + 1;

Else

| B = B + 1;

}

printf("case: ", i);

printf("the number of present students is: ", B);

printf("the number of absent students is: ", C);

IF (B ≥ A && C < S)

printf("Session Valid");

IF (B < A && C ≥ S)

printf("Session cancelled");

}

}

P = B + C;

printf("the number of total processed students is: ", P);

return 0;

{

Copy 13

```
#include <stdio.h>
int main()
{
    int N, A, S, C = 0, B = 0, i, P, 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);
    while (i <= N && i <S)
    {
        // Switch case (i) printf("Enter the number of attended sessions of student: %d", i);
        scanf("%d", &X);
        if (X <A)
        {
            C = C + 1;
        }

        else
        {
            B = B + 1;
        }

        printf("the number of present students is: %d", B);
        printf("the number of absent students is: %d", C);
        if (B>= A && C <S)
        {
            printf("session Valid");
        }

        if (B <A && C>= S)
        {
            printf("session cancelled");
        }

    }

    P = B + C;
    printf("the number of total processed students is: %d", P);
    return 0;
}
```

Analyse :

Algorithmique :

- Commentaire `// Switch case`.
- Boucle `while`. `i` non initialisé. `i` jamais incrémenté (boucle infinie).
- Code logique interne correct.

NOTE FINALE : 06 / 20

Feedback :

- **Appréciation globale : Insuffisant.** Boucle infinie.
-

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

```
#include <stdio.h>
#include <stdlib.h>
#include <bool.h>
int main() {
    int N, A, S, X, i=1, present_a=0,
        absent_a=0, processed_a=0;
    present_a=0, absent_a=0, processed_a=0;
    bool session_cancelled;
    printf("Enter number of registered students : ");
    scanf("%d", &N);
    printf("Enter the minimum attendance required : ");
    scanf("%d", &A);
    printf("Enter the absence threshold : ");
    scanf("%d", &S);
    while (i <= N && absent_a != S) {
        printf("Enter number of attended sessions for student %d : ", i);
        scanf("%d", &X);
        if (X < A) {
            absent_a += 1;
        }
        else {
            present_a += 1;
        }
        processed_a += 1;
        if (absent_a == S) {
            session_cancelled = 1;
        }
        i++;
        printf("total processed students : %d", processed_a);
        printf("present students : %d", present_a);
        printf("absent students : %d", absent_a);
        if (session_cancelled) {
            printf("session cancelled");
        }
        else {
            printf("session valid");
        }
    }
}
```

```

    else {
        present_a += 1;
    }
    printf("present students : %d", present_a);
    printf("absent students : %d", absent_a);
    processed_a += 1;
    if (absent_a == S) {
        session_cancelled = 1;
    }
    i++;
    printf("total processed students : %d", processed_a);
    printf("present students : %d", present_a);
    printf("absent students : %d", absent_a);
    if (session_cancelled) {
        printf("session cancelled");
    }
    else {
        printf("session valid");
    }
}
}
```

Copy 14

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
int main()
{
    int N, A, S, X, i = 1, present_a = 0, absent_a = 0;
    // bool session_cancelled;
    printf("Enter number of registered students");
    scanf("%d", &N);
    printf("Enter the minimum attendance required");
    scanf("%d", &A);
    printf("Enter the absence threshold");
    scanf("%d", &S);
    while (i <= N && absent_a != S)
    {
        printf("Enter number of attended sessions for student %d", i);
        scanf("%d", &X);
        if (X < A)
        {
            absent_a += 1;
        }
        else
        {
            present_a += 1;
        }

        printf("present students: %d", present_a);
        printf("absent students: %d", absent_a);
        if (absent_a == S)
        {
            // session_cancelled = 1;
            printf("total processed student %d", processed_a);
            printf("session cancelled");
        }
        else
        {
            printf("session valid");
        }
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Boucle `while` correcte (`absent_a != S`).
- Logique correcte.
- Affichage propre.
- Petit souci : `i` n'est incrémenté nulle part. Boucle infinie sur le 1er étudiant.

NOTE FINALE : 12 / 20

Feedback :

- **Appréciation globale : Moyen.** Oubli d'incrémenter le compteur de boucle.
-

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, N, A, S, X, Z = 0, V = 0;
    printf("Enter the number of student");
    scanf("%d", &N);
    printf("Enter the minimum attendance required");
    scanf("%d", &A);
    printf("Enter the absence threshold");
    scanf("%d", &S);

    for (i = 1, i <= N, i++) {
        while (Z < S) {
            printf("Enter the number of attender sessions %d: ", i);
            scanf("%d", &X);

            if (X > A) {
                V = V + 1;
                printf("the student %d is present", i);
            }
            else (X < A) {
                Z = Z + 1;
                printf("the student %d is absent", i);
            }
        }
    }

    printf("The number of student absent is: %d", Z);
    printf("The number of student present is: %d", V);

    if (V > A) {
        printf("Session valid");
    }
    else (V < A) {
        printf("Session cancelled");
    }

    return 0;
}

```

```
#include <stdio.h>
int main()
{
    int i, N, A, S, X, Z = 0, V = 0;
    printf("Enter the number of student");
    scanf("%d", &N);
    printf("Enter the minimum attendance required");
    scanf("%d", &A);
    printf("Enter the absence threshold");
    scanf("%d", &S);
    for (i = 1; i <= N; i++)
    {
        while (Z < S)
        {
            printf("Enter the number of attender sessions %d: ", i);
            scanf("%d", &X);
            if (X > A)
            {
                V = V + 1;
                printf("the student %d is present", i);
            }

            else if (X < A)
            {
                Z = Z + 1;
                printf("the student %d is absent", i);
            }
        }
    }

    printf("the number of student absent is: %d", Z);
    printf("the number of student present is: %d", V);
    if (V > A)
    {
        printf("session valid");
    }

    else
    {
        printf("session cancelled");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Double boucle for (étudiants) + while (tant que $Z < S$??).
- Le while à l'intérieur demande de saisir des sessions tant que le seuil global d'absence n'est pas atteint ?
- Confusion entre seuil global et saisie individuelle.
- Si un étudiant a $X > A$, V augmente. Le while ne s'arrête que si $Z < S$. Si l'étudiant est présent, Z ne change pas -> Boucle infinie sur la saisie du même étudiant ?

NOTE FINALE : 08 / 20

Feedback :

- **Appréciation globale : Fragile.** Risque de boucle infinie.
-

```

#include <stdio.h>
int main() {
    int i, N, S, X, A;
    while
    scanf("%d%d%d%d%d", &i & N & S & X & A);
    printf("N, S, X, A");
    for while (i = 1, i ≤ N; i++) {
        printf("the number of Student is: %d N);
        if (X < A) {
            printf("present Student");
        } else {
            printf("absent Student");
        }
        if (absent student = S) {
            printf("Session cancelled");
        } else {
            printf("Session valid");
        }
    }
    return 0
}

```

Copy 16

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

        else
        {
            printf("absent student");
        }

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

        else
        {
            printf("Session valid");
        }

    }

    return 0;
}
```

Analyse :

Algorithmique :

- Lecture groupée bizarre `scanf ( ..., &i ... )`. Écrase `i`.
- Boucle `for`.
- Ne lit pas `X` dans la boucle. Utilise la valeur lue au début (n'importe quoi).

NOTE FINALE : 06 / 20

Feedback :

- **Appréciation globale : Insuffisant.**
-

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

#include <Stdio.h>

int main() {

int N, A, S, absent = 0, present = 0, i, x;

printf("enter The total number of Student");

scanf("%d", &N);

printf("enter The Minimum attendance Required");

scanf("%d", &A);

printf("enter The absence Threshold");

scanf("%d", &S);

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

{ printf("There is Till now: \n %d present counted \n %d absent counted", present, absent);

printf("This is student number: %d, How Many sessions He attended?", i);

scanf("%d", &x);

if(x < A) { absent = absent + 1 }

else { present = present + 1 }

if(absent == S) { i = N + 1; }

}

printf("The Total processed Student are: %d", i);

printf("In the present student are: %d \n", present);

printf("The number of absent is: %d \n", absent);

if(absent == S) { printf("The session is canceled"); }

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

Copy 17

```
#include <stdio.h>
int main()
{
    int N, A, S, absent = 0, present = 0, i, x;
    printf("enter the Total number of Student");
    scanf("%d", &N);
    printf("enter The Minimum attendance Required");
    scanf("%d", &A);
    printf("enter The absence Thershold");
    scanf("%d", &S);
    for (i = 1; i <= N; i++)
    {
        printf("There is Till now: %d present counted \n %d absent counted", present, absent);
        printf("This is student number: %d, How Many sessions He attended?", i);
        scanf("%d", &x);
        if (x < A)
        {
            absent = absent + 1;
        }

        else
        {
            present = present + 1;
        }

        if (absent == S)
        {
            i = N + 1;
        }

    }

    printf("The Total processed student are: %d", i);
    printf("The present student are: %d", present);
    printf("The number of absent is: %d", absent);
    if (absent == S)
    {
        printf("The session is canceled");
    }

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

}
```

Analyse :

Algorithmique :

- Boucle for.
- Arrêt if (absent == S) i = N + 1. Sortie de boucle correcte (hack).
- Logique correcte.
- Variable x lue correctement.

NOTE FINALE : 17 / 20

Feedback :

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

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

```

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

```

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

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

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

        else
        {
            printf("session cancelled");
        }

    }

    return 0;
}
```

Analyse :

Algorithmique :

- Lecture confuse (scanf successifs écrasant les vars).
- Utilise x non lu dans le `if`.

NOTE FINALE : 05 / 20

Feedback :

- **Appréciation globale : Insuffisant.**
-

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

~~stdio~~ include <stdio.h>

int N, A, S, X;

int Z = 0, M = 0, i = 1

int main {

printf("العدد الإجمالي للطلبة المسجلين");

scanf("%d", &N);

printf("الحد الأدنى للداخيل");

scanf("%d", &A);

printf("عتبة الضيافة المسموح بها");

scanf("%d", &S);

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

~~while~~ printf("عدد الصف التي دفعتها المال");

scanf("%d", &X);

if (X < A)

{ Z = Z + 1

else

M = M + 1

}

if (Z < S)

{ printf("%d", Z);

printf("%d", "عدد الطلبة الفاضلين هو");

else

i = N + 1

~~while~~ printf("الامتحان ملغى");

{

~~(end for)~~ if (i == N){ ~~while~~ printf("الامتحان صالح"); }

end for

{

return 0;

{

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

        else
        {
            M = M + 1;
        }
    }

    if (Z == 5)
    {
        printf("%d ■■■ ■■■■■■ ■■■■■■■■ ■■", Z);
        printf("%d ■■■ ■■■■■■ ■■■■■■■■ ■■", M);
    }

    else
    {
        i = N + 1;
        printf("■■■■■■■■ ■■■■");
    }

    if (i == N)
    {
        printf("■■■■■■■■ ■■■■");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Commentaires en Arabe/Français.
- Lectures correctes.
- Compteurs Z, M.
- Condition finale `if (Z == 5)`. Valeur magique 5 ? (Peut-être S vaut 5 dans son esprit ?).
- Condition boucle sur N uniquement.

NOTE FINALE : 13 / 20

Feedback :

- **Appréciation globale : Moyen.** Constante magique inexpliquée.
-

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 = 1, p, w;
    printf("Enter N");
    scanf("%d", &N);
    printf("Enter S");
    scanf("%d", &S); printf("Enter A"); scanf("%d", &A);
    while (i < N, i != S) {
        printf("Enter x");
        scanf("%d", &x);
        if (x < A) {
            p = p + x;
            i++;
            printf("عدد الطلبة المبرزين");
        }
        else
            w = N - p;
        i++;
        printf("عدد الطلبة العاديين");
    }
    if (p >= A) {
        printf("الإمتان صالح");
    }
    else
        printf("الإمتان ملغى");
    return 0;
}

```

```
#include <stdio.h>
int main()
{
    int N, A, S, x, i = 1, p = 0, a = 0;
    printf("Enter N");
    scanf("%d", &N);
    printf("Enter S");
    scanf("%d", &S);
    printf("Enter A");
    scanf("%d", &A);
    while (i <N && i != S)
    {
        printf("Enter x");
        scanf("%d", &x);
        if (x <A)
        {
            p = p + i;
            i++;
            printf("p = %d", p);
        }

        else
        {
            a = N - p;
            i++;
            printf("a = %d", a);
        }

    }

    if (p>= A)
    {
        printf("■■■■■■■■■■ ■■■■");
    }

    else
    {
        printf("■■■■■■■■■■ ■■■■");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Condition `while (i < N && i != S)`. Utilise l'indice `i` au lieu du compteur d'absents. Arrêt prématuré si on arrive à l'étudiant numéro `S`.
- Calcul `p = p + i`. Somme des indices ? Faux.

NOTE FINALE : 08 / 20

Feedback :

- **Appréciation globale : Fragile.**
-

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

```

#include <stdio.h>
#include <stdlib.h>
int main() {
    int N, A, S, X, K, n, i = 0, p;
    printf("Enter the total number of registered students");
    scanf("%d", &N);
    printf("Enter the minimum attendance required");
    scanf("%d", &A);
    printf("Enter absence threshold");
    scanf("%d", &S);
    while (N > 0) {
        printf("Enter the number of student");
        scanf("%d", &K);
        printf("Enter the number of attended sessions: X");
        scanf("%d", &X);
        if (X < A) {
            printf("The student is absent");
        } else (X > A) {
            printf("The student is present");
        }
        for (n = i + 1, i++) for (n = i + 1, i++);
        printf("Enter the total number of present students");
        scanf("%d", &n);
        for (p = i + 1, i++);
        printf("Enter the total number of absent students");
        scanf("%d", &p);
        if (p > n) {
            printf("Session cancelled");
        } else if (n > p) {
            printf("Session Valid");
        }
        return 0;
    }
}

```

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

        else if (X > A)
        {
            printf("The student is present");
        }

        for (n = i + 1; i++)
        {
            printf("Enter the total number of present students");
            scanf("%d", &n);
            for (p = i + s; i++)
            {
                printf("Enter the total number of absent students");
                scanf("%d", &p);
                if (p > n)
                {
                    printf("session cancelled");
                }

                else if (n > p)
                {
                    printf("session Valid");
                }
            }
        }
    }
}
```

```
}  
    return 0;  
}
```

Analyse :

Algorithmique :

- Boucle `while` (`N > 0`). Boucle infinie (N ne change pas).
- Boucles `for` imbriquées bizarres.
- Demande de saisir le nombre d'absents ? (Contresens).

NOTE FINALE : 04 / 20

Feedback :

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

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

```

#include <stdio.h>

int main() {
    int N, A, S, X, absence = 0, attended = 0, random = 1;
    printf("enter the number of registered students");
    scanf("%d", &N);
    printf("enter the number of minimum attendance");
    scanf("%d", &A);
    printf("enter the number of absence threshold");
    scanf("%d", &S);
    for (i = 1; i <= N && absence <= S; i++) {
        printf("enter the number of attended sessions for student %d \n", i);
        scanf("%d", &X);
        if (X < A)
            random = 0;
        if ((random))
            printf("the student %d has attended", i);
            attended = attended + 1;
        else
            printf("the student %d is absent");
            absence = absence + 1;

        printf("attended = %d, absent = %d \n", attended, absence);
    }
    printf("Total attended = %d \n Total absence = %d \n",
        attended, absence);
    if (absence == S) printf("session invalid");
    else printf("session valid");
    return 0;
}

```

out put on each step:
 student number : i
 attendance = attended
 absence = absence

final output:
 Total attendance
 Total absence
 session status

```
#include <stdio.h>
int main()
{
    int N, A, 5, X, absence = 0, attended = 0, random = 1;
    printf("enter the number of registered students");
    scanf("%d", &N);
    printf("enter the number of minimum attendance");
    scanf("%d", &A);
    printf("enter the number of absence threshold");
    scanf("%d", &S);
    for (i = 1; i <= N && absence <= S; i++)
    {
        printf("enter the number of attended sessions for student %d\n", i);
        scanf("%d", &X);
        if (X < A)
        {
            if (random)
            {
                printf("the student %d has attended", i);
                attended = attended + 1;
            }

            else
            {
                printf("the student %d is absent", i);
                absence = absence + 1;
            }
        }
    }

    printf("attended = %d, absent = %d", attended, absence);
    printf("Total attended : %d \n Total absence : %d \n", attended, absence);
    if (absence == S)
    {
        printf("session invalid");
    }

    else
    {
        printf("session valid");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Variable 5 en nom de variable ? (`int N, A, 5...`) Syntaxe illégale.
- Variable `random` utilisée non init (non, init à 1).
- `if (X < A) : if (random) ->` condition toujours vraie. Compte les absents comme présents ("attended") ? Logique inversée et étrange.
- Tente de simuler de l'aléatoire ?

NOTE FINALE : 06 / 20

Feedback :

- **Appréciation globale : Insuffisant.** Code confus.
-

Copy number : 24

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;
    Printf("Enter X, N");
    Scanf("%d", &X);
    Scanf("%d", &A);
    Scanf("%d %d", &N, &S);
    while (X < A) {
        S = S + 1;
        Printf("%d", S);
    }
    P = N - S;
    Printf("%d", P);
    if (P <= S) {
        Printf("Session Cancelled");
    }
    else {
        Printf("Session Valid");
    }
    return 0;
}
```

```
#include <stdio.h>
int main()
{
    int N, A, S, X, P;
    printf("Enter N");
    scanf("%d", &X);
    scanf("%d", &A);
    scanf("%d %d", &N, &S);
    while (X < A)
    {
        S = S + 1;
        printf("%d", S);
    }

    P = N - S;
    printf("P = %d", P);
    if (P <= S)
    {
        printf("Session cancelled");
    }

    else
    {
        printf("Session Valid");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Lecture chaotique. Ecrasement variables.
- Boucle `while (X < A)`. Condition sur variable d'entrée non modifiée -> Boucle infinie ou jamais exécutée.
- Incrémente S ?

NOTE FINALE : 03 / 20

Feedback :

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

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

```

#include <stdio.h>
int main() {
    int A, N, S; // integers;
    printf("Examination Attendance Monitoring");
    scanf("%d %d %d", &A, &N, &S);
    while (N != A) {
        scanf("%d", &N);
        if (N < A) {
            i = i + 1;
            printf("The student is absent");
        }
        else (N > A) {
            i = i - 1;
            printf("The student is present");
        }
    }
    printf("Student = N");
    printf("Student = A");
    if (N > S) {
        printf("session valid");
    }
    else {
        printf("session cancelled");
    }
    return 0;
}

```

```
#include <stdio.h>
int main()
{
    int A, N, S;
    // integers printf("Examination Attendance Monitoring");
    scanf("%d %d %d", &A, &N, &S);
    while (x != A)
    {
        scanf("%d", &x);
        if (x < A)
        {
            i = i + 1;
            printf("The student is absent");
        }

        else if (n > A)
        {
            i = i - 1;
            printf("The student is present");
        }

    }

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

    else
    {
        printf("session cancelled");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Boucle `while (x != A)`. Condition d'arrêt étrange (tant que la saisie != seuil ?).
- Incrémentation `i` dans les deux sens (+1 ou -1).
- Affichage final douteux.

NOTE FINALE : 05 / 20

Feedback :

- **Appréciation globale : Insuffisant.**