

```

#include <stdio.h>

int main() {

    int N, A, S, X;

    printf("entre number of N");
    scanf("%d", &N);
    printf("entre number of x");
    scanf("%d", &X);

    if (X >= A) {
        printf("the student is Present");
    }
    if (X < A) {
        printf("the student is absent");
    }

    for (i = 1; i < N; i++) {
        // while (S > T) {
        printf("how many student %d com?", X);
        scanf("%d", &X);
        if (X >= A) {
            R++;
        }
        else {
            T++;
        }
    }
    if (S > T) {

```

```

        printf("nopt student %d", A);
    }
    else {
        printf("semanvelist");
    }
    else {
        printf("semanvelist counselled");
        return 0;
    }
}

```

Copy 1

```
#include <stdio.h>
int main ( )
{
    int N, A, S, X;
    printf ( " entre nenbeer of N " );
    scanf ( " %d " , N);
    printf ( " entre nenbeer of X " );
    scanf ( " %d " , x ) i if ( x> = A )
    {
        printf ( " the Student is Present " );
    }

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

    for ( i = 1 ; i <N ; i ++ )
    {
        int i ;
        while ( s> T )
        {
            printf ( " how monvy student %d com ? " , X ) ;
            scanf ( " %d " & X ) " if ( X> = A )
            {
                R + + ;
            }

            else
            {
                T + + ;
            }

        }

        if ( S> T )
        {
            printf ( " psopt student %d " , A );
            else . print ( " seman valid " );
            else
            {
                printf ( " seman cancelled " ) ;
                return 0 ;
            }
        }
    }
}
```

Analyse :

Algorithmique :

- Lecture N, X (avant boucle). X lu hors boucle.
- Boucle for avec while imbriqué ? while (s > T). s minuscule (S majuscule ailleurs). T non initialisé.
- Redéclare int i dans la boucle.
- Syntaxe très approximative (while, nenbeer, if (S > T) mal placé).

NOTE FINALE : 03 / 20

Feedback :

- **Appréciation globale : Très Insuffisant.** Syntaxe et logique chaotiques.
-

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

```
printf
i++;
printf("the number of student is
%d\n", i);
printf("the present student are:
%d\n", present student);
printf("the absent student are:
%d\n", absent student);
}
printf("the total number of
absent present is: %d\n",
present student);
printf("the total number of
absent is: %d\n",
absent student);
if (absent student < S)
{ printf("session ended");
}
else {
    printf("session cancelled");
}
return 0;
}
```

Copy 2

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

        else
        {
            print F ("The student is present.");
            present student ++;
        }

        i++;
        print F ("the number of student is %d\n", i);
        print F ("the present student are: %d\n", present student);
        print F ("the absent student are: %d\n", absent student);
    }

    print fl ("the totale numbre of present is : %d\n", present student ) ;
    print F ("the totale numbre of absent is : %d\n" absent student);
    if (absent student> S)
    {
        print F ("session valid");
    }

    else
    {
        print F ("session cancelled");
        return 0;
    }
}
```


Analyse :

Algorithmique :

- Initialisation variables verbeuse mais correcte.
- Boucle `while` avec condition d'arrêt correcte (`i <= N && S <= absent`). Attention, énoncé dit arrêt si `absent >= S` (donc pas strictement `<` ?). Ici `S <= absent` signifie "tant que S est inf ou égal à absent" ? Non, l'étudiant veut dire "tant que absent n'a pas atteint S". Si `absent < S`, la boucle continue. Ici écrit `S <= absent`, donc si `S=3` et `absent=0`, boucle ne s'exécute pas. Condition inversée.
- Logique interne OK.

NOTE FINALE : 10 / 20

Feedback :

- **Appréciation globale : Moyen.** Erreur logique majeure dans la condition du `while` (inversée).
-

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 x, N, A, S; i = 0; j = 0;
    scanf("%d", &x);
    if (x < A) {
        printf("The student is absent");
    }
    else {
        printf("The student is present");
        scanf("Enter the number of present students");
        scanf("%d", &i);
        scanf("Enter the number of present student");
        for (i = 1; i <= N; i++) {
            printf("Enter the number of present students is: i");
            scanf("Enter the number of absent students");
            scanf("%d", &j);
            for (j = 1; j <= N; j++) {
                printf("The number of absent student is: j");
                scanf("Enter the number of absent student");
            }
            printf("Total processed students is: i, j, i+j");
            if (j+i != N && i != S) {
                printf("The session valid");
            }
            else {
                printf("The session cancelled");
            }
        }
    }
}

```

Copy 3

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

            Printf ( " Total processed studens is : % d , % d , % d " , j , i , i + j )
            if ( j + i != N && j != S )
            {
                Printf ( " The session valid " );
            }

            else
            {
                ( " The session cancelled " );
            }
        }
    }
}
```

Analyse :

Algorithmique :

- Lecture confuse (n=0; j=0).
- Boucles imbriquées for (ii et j).
- Mélange les compteurs.
- Syntaxe Scanf majuscule.
- Affichage final incohérent.

NOTE FINALE : 04 / 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

```

Exo 8
#include <stdio.h>
int main() {
    int N, A, S;
    char *valid, *cancelled;
    printf("Enter the number of registered students : N");
    scanf("%d", &N);
    printf("Enter the min attendance required : A");
    scanf("%d", &A);
    printf("Enter the absence threshold : S");
    for (i = A; i < A; i++) {
        printf("Enter x");
        scanf("%d", &x);
        if (x < A) {
            printf("the student is absent");
        }
        Else {
            printf("the student is present");
        }
        if (the absent students = S) {
            printf("Session cancelled");
        }
    }
}

```

```

else {
    printf("Session Valid");
}
return 0;
}

```

Copy 4

```
Exo 8 include ( <stdio.h> ) int main ( )
{
    int N, A, S, char "valid", "cancelled";
    Printf ( " Enter the number of registered students : N " );
    Scanf ( " %d ", & N );
    Printf ( " Enter the min attendance required " : A );
    Scanf ( " %d ", & A );
    Printf ( " Enter the absence threshold " : S );
    for ( i = A ; i <A ; i ++ )
    {
        Printf ( " enter x " );
        Scanf ( " %d ", & x );
        if ( x <A )
        {
            Printf ( " the student is absent " );
        }

        Else
        {
            Printf ( " the student is present " );
        }
    }

    if ( the absent sutudents = S )
    {
        Printf ( " Session cancelled " );
        else
        {
            Printf ( " Session valid " );
        }
    }

    return 0;
}
}
```

Analyse :

Algorithmique :

- Boucle for (i=A; i<A; i++). Ne s'exécute jamais.
- Variables char "valid".
- Syntaxe include (<stdio.h >.

NOTE FINALE : 02 / 20

Feedback :

- **Appréciation globale : Très Insuffisant.** Programme vide logiquement.
-

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, n;
    printf("%d", &N);
    printf("the total number of registered students");
    scanf("%d", &A);
    printf("the minimum attendances required");
    scanf("%d", &S);
    printf("the absence threshold");
    do {
        scanf("%d", &n);
        printf("the number of attended sessions");
        if (n < A) {
            printf("the student is absent");
        } else {
            printf("the student is present");
        }
    } while ((n == N) || (n == S));
    return 0;
}

```

Copy 5

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

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

    }

    whill ((n = N) || (n = S));
    return 0;
}
```

Analyse :

Algorithmique :

- Boucle `Do . . . while`. Syntaxe `while`.
- Condition `((n=N) || (n=S))`. Utilise `=` (affectation) au lieu de `==`.
- Logique interne OK.

NOTE FINALE : 08 / 20

Feedback :

- **Appréciation globale : Insuffisant.** Erreurs de syntaxe (`while`, `=`) bloquantes.
-

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, B; i=1 i=1;
    printf("Enter total number of registered student N");
    scanf("%d", &N);
    printf("Enter minimum attendance required A");
    scanf("%d", &A);
    printf("Enter Absence threshold S");
    scanf("%d", &S);
    printf("Enter the number of attend session n");
    scanf("%d", &n);
    printf("Enter number of absent present session B");
    scanf("%d", &B);
    while (i <= N && B < S) {
        printf("Enter x member present");
        if (x (x < A))
            printf("Absence");
        else
            printf("Present");
        i++;
    }
    printf("present = %d\n", present);
    Absence = N - Present;
    printf("Absence = %d\n", Absence);

    int int B, P;
    printf("Enter B number of Absence");
    scanf("%d", &B);
    printf("Enter P number of present");
    scanf("%d", &P);
    if (P > B)
        printf("session valid");
    else
        printf("session canceled");
    return 0;
}

```

Copy 6

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

    printf ( "present = %d" , present );
    Absence = N - Present ;
    printf ( "Absence = %d", Absence );
    int B, P;
    printf ( "Enter B number of Absence" );
    Scanf ( "%d", & B );
    printf ( "Enter P number of present" );
    Scanf ( "%d", & P );
    if ( P> B ) printf ( "session valid" );
    else printf ( "session canceled" );
    return 0;
}
```

Analyse :

Algorithmique :

- Boucle `while` correcte.
- Lit `n` et `B` avant la boucle ?
- Dans la boucle, lit `x` ("member present"?).
- Condition interne `if (x < A)` OK.
- Calculs finaux un peu brouillons (`Absence = N - Present`). Logique valide si `Present` est correct.

NOTE FINALE : 11 / 20

Feedback :

- **Appréciation globale : Passable.** Logique à peu près tenue malgré des maladresses.
-

Copy number :

7

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

in cloud « studio.h ».

int main.

printf.

printf « entre A »

int

printf « entre B ».

int
printf « entre C ».

return 0.

Copy 7

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

Analyse :

Algorithmique :

- Pseudo-code très vague. `printf`, `in clond`. Rien de compilable.

NOTE FINALE : 00 / 20

Feedback :

- **Appréciation globale : Nul.**
-

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

Copy 8

```
#include <stdio.h>
int main ( )
{
    int N, A, S, x, i ;
    printf ( "enter the student number" ) ;
    S = 0 ;
    n = 0 ;
    for ( i = 1 ; i <= N ; i ++ )
    {
        if ( x < A )
        {
            S = S + i ;
        }

    }

    printf ( "Enter the minimum attendance reqewred" ) ;
    scanf ( "%d", & A );
    if ( S > A )
    {
        printf ( "session cancelled" ) ;
    }

    else
    {
        printf ( "Session valid" ) ;
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Lecture manquante pour N (utilisé dans for).
- Logique : $S = S + i$? Somme les indices ?
- Lit A à la fin.
- Aucun sens par rapport à l'énoncé.

NOTE FINALE : 01 / 20

Feedback :

- **Appréciation globale : Hors Sujet.**
-

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

```

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

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

    present_students = ++present;
    absent_students = ++absent;

    printf("present students", present_students);
    printf("absent students", absent_students);
    for (present_students >= N & & absent_students < s); {
        printf("sessions valide");
    }
    while (present_students < N & & absent_students > s); {
        printf("session cancelled");
    }
    return 0;
}

```

Copy 9

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

    else printf ("the student is present", present);
    present-students = ++ present absent-students = ++ absent;
    printf ("present students", present-students);
    printf ("absent students", absent-students);
    for (present-students> N && absent-students <S)
    {
        printf ("session valicle");
    }

    while (present-students <N && absent-students> S)
    {
        printf ("session cancelled");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Pas de boucle principale englobante (juste séquentiel).
- `if (x < A)` une seule fois.
- Incrémentation bizarre `present-students = ++ present`.
- Utilise `for` comme un `if` à la fin.

NOTE FINALE : 05 / 20

Feedback :

- **Appréciation globale : Insuffisant.** Ne traite pas N étudiants.
-

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 N")
    scanf("N is total number of
           registered students")
    printf("Enter the A")
    scanf("A is minimum attendance required")
    printf("Enter the S")
    scanf("S is absence threshold")
    printf("Enter the x")
    scanf("x it is the number of attended sessions")
    if ("x < A the student is absence") {
        printf("Enter the students present, the student's absence")
        3
    }
    return 0;
}

```

Copy 10

```
#include <stdio.h>
int main ( )
{
    int N, A, S, X ;
    printf ( "Enter the N" ) Scanf ( "N is total number of registered students" ) printf (
    {
        printf ( "Enter the Students present, the students absence" )
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Pas de boucle.
- Juste des `printf` et `scanf` séquentiels.
- Mélange chaînes et code.

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>
#include <stdlib.h>
int main() {
    int N, A, S;
    printf("Enter the total number of registered students", N);
    scanf("%d", &N);
    printf("Enter the minimum attendance required", A);
    scanf("%d", &A);
    printf("Enter the absence threshold", S);
    scanf("%d", &S);
    x = Read the number of attended session;
    if (x=0; x<A; x++);
    printf("The student is considered absent");
    else,
        printf("The student is present");
}
{ if ("all students are processed or the number of absent students reaches");
    printf("Simulation stops");
}
3
printf("Enter the total processed students", N);
printf("Enter the number of absent students && present students");
if (N=0, N<S, N++);
printf("The session cancelled");
else,
    printf("The session valid");
}
return 0;

```

Copy 11

```
#include <stdio.h>
#include <stdlib.h>
int main ( )
{
    int N, A, S ;
    Printf ( " Enter the total number of registred students ", N ) ;
    Scanf ( " %d ", &N ) ;
    Printf ( " Enter the minimum attendance required ", A ) ;
    Scanf ( " %d ", &A ) ;
    Print ( " Enter the absence threshold ", S ) ;
    Scanf ( " %d ", &S ) ;
    x = Read the number of attended sessions ;
    if ( x = 0 ; x <A ; x ++ ) ;
    Print f ( " the student is considred absent " ) ;
    else , Print f ( " the student is present " ) ;
}

{
    if ( " all students are processed or the number of absent students reaches " ) ;
    Print f ( " Simulation stops " ) ;
}

Print f ( " Enter the total processed students ", N ) ;
Print f ( " Enter the number of absent students & present students " ) ;
if ( N = 0, N <S, N ++ ) ;
Print f ( " The session conclled " ) ;
else , Print f ( " The session valid " ) ;
}

return 0 ;
```

Analyse :

Algorithmique :

- Syntaxe `Print f`.
- `x = Read ....` Pseudo-code.
- Sépare `{` et `}` de manière aléatoire.

NOTE FINALE : 02 / 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;
    int s, x;
int x = N/s;

    printf("enter the total number of registered students:");
    scanf("%d", &N);

    printf("enter the minimum attendance required:");
    scanf("%d", &A);

    printf("enter the absence threshold:");
    scanf("%d", &s);

    if (x < A) {
        printf("Absent");
    }
    else { printf("Present"); }

    for (int x = N/s; x < A; x++) {
        printf("Present");
        N++;
    }

    return 0;
}

```

Copy 12

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

    else
    {
        Printf ( " Present " ) ;
    }

    for ( int x = N/S ; x <A ; x ++ )
    {
        Printf ( " Present " ) ;
        N ++ ;
    }

    return return 0 ;
```

Analyse :

Algorithmique :

- Boucle for (`int x = N/S ...`). Rien à voir.
- `return return 0.`

NOTE FINALE : 03 / 20

Feedback :

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

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, S;
    int x;
    Printf("enter the number of attended session");
    scanf("%d", &x);
    if (x < A) {
        Printf("absent student");
    } else {
        Print("present student");
    }
}
```

Copy 13

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

    else
    {
        Print ( " present student " ) ;
    }
}
```

Analyse :

Algorithmique :

- Pas de boucle. Traitement d'un seul cas.

NOTE FINALE : 03 / 20

Feedback :

- **Appréciation globale : Insuffisant.** Incomplet.
-

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

```

Copy 14

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

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

        printf ( " present students " ) ;
        printf ( " absent students " ) ;
        i ++
    }

    if (absent)
```

Analyse :

Algorithmique :

- Boucle `while` correcte.
- Logique interne correcte.
- Compteurs OK.
- Condition fin incomplet `if (absent)`. Code coupé fin ?
- Dans l'ensemble, début solide.

NOTE FINALE : 13 / 20

Feedback :

- **Appréciation globale : Moyen.** Bon début, mais fin manquante/incorrecte.
-

```

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

```

Copy 15

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

        else
        {
            if ( A <= X )
            {
                printf ( " student is present /n " ) ;
                p = p + 1 ;
            }

            printf ( " absent students % d ", b ) ;
            printf ( " present students % d ", p ) ;
            if ( i == b == s )
            {
                printf ( " the number of absent reaches S : % d ", S ) ;
            }

            else
            {
                printf ( " session canceled " ) ;
                printf ( " session valid " ) ;
            }

            return 0 ;
        }
    }
}
```

Analyse :

Algorithmique :

- Boucle for avec condition complexe `i <= (b == S)`. `(b==S)` vaut 0 ou 1. Donc boucle s'arrête tout de suite.
- Logique interne `A > X` OK.
- Affichage à chaque itération.

NOTE FINALE : 09 / 20

Feedback :

- **Appréciation globale : Moyen.** Erreur dans la condition 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 N, A, S, P, a, n, i, X;

    printf("Enter the Total number registered Student: ");
    scanf("%d", &N);
    printf("Enter The minimum attendance required: ");
    scanf("%d", &A);
    printf("The Absence threshold: ");
    scanf("%d", &S);

    for (i = 0; i < N; i++)
        if (X < A)
            printf("The Student is Absence: ");
        else
            printf("The Student is Present: ");

    P = N - (X < A);
    printf("The Number of Students presents is: %d", P);
    a = N - (P >= S);
    printf("The Number of Students absent is %d", a);
    n = N - (P + a);
    printf("The Number of Total processed Students is %d", n);
    if (n <= S)
        printf("The Session is Valid: ");
    else
        printf("The Session is Cancelled: ");

    return 0;
}

```

Copy 16

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

Analyse :

Algorithmique :

- Boucle For (`i=0; n; i++`). `n` non initialisé.
- Calculs via booléens `P = N - (X < A)`. Astucieux si maîtrisé, mais ici `X` n'est pas lu dans la boucle (ou pas de boucle réelle).
- Syntaxe `Print f`.

NOTE FINALE : 06 / 20

Feedback :

- **Appréciation globale : Insuffisant.** Approche confuse.
-

```

int main() { input, output;
  int ( N, A, s, x, i ); [
  printf( N: A < x, i );
  scanf: ( the s ) :
  printf ( N: x > A )
  scanf ( N, x ) ;
  if ( n = < i = 6 )
    continue
  if ( all n = s )
    stop

```

Copy 17

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

Analyse :

Algorithmique :

- Pseudo-code incompréhensible (printf, in clond, ??).

NOTE FINALE : 00 / 20

Feedback :

- **Appréciation globale : Nul.**
-

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 x;
    int i = 0;
    Present = 0, Absent = 0;

    printf ("Enter the Total num of student N: ");
    scanf ("%d", &N);

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

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

    while (i < N && absent < S) {
        printf ("Enter attended sessions for student %d ", i + 1);
        scanf ("%d", &x);

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

        printf ("step %d -> Present: %d | Absent: %d", i, present, absent);
        printf ("\n Final results: \n");
        printf ("Total processed students: %d", i);
        printf ("Present students: %d", Present);
        printf ("Absent students: %d", Absent);
    }

```

return 0;

{

Copy 18

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

        else
        {
            Print f ( " the student is present " ) ;
            Print f ( " Step d -> present : % d | Absent : % d ", i, present, absent ) ;
            Print f ( " Final results : \n " ) ;
            Print f ( " Total Processed students : % d ", i ) ;
            Print f ( " Present students : % d ", Present ) ;
            Print f ( " Absent students : % d ", Absent ) ;
            return 0 ;
        }
    }
}
```

Analyse :

Algorithmique :

- Boucle `while` correcte.
- Lecture `X` dans la boucle.
- Logique `if (x < A)` OK.
- Affichage intermédiaire OK.
- Tout est mis dans le `else` du `if (x < A)` ? Indentation suggère que la fin de boucle est mal placée. Il manque une accolade fermante pour le `else` avant `}` de boucle. La validation finale est Absente.

NOTE FINALE : 11 / 20

Feedback :

- **Appréciation globale : Passable.** Structure presque correcte, erreurs d'accolades.
-

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, a, S; int y=0, z=0, x;
    printf("Enter A: ");
    scanf("%d", &A);
    printf("Enter S: ");
    scanf("%d", &S);
    printf("Enter x: ");
    scanf("%d", &x);
    for (i=0; (i < A) || (i < S); i++) {
        printf("number of attended session student %d, i+1);
        scanf("%d", &x);
        if (x < A) { printf("student %d absent", i+1);
            y++; }
        else { printf("student %d present", i+1);
            z++; }
    }
    printf("Students present is %d", z);
    printf("Students absent is %d", y);
    if (z < y) { printf("session valid"); }
    else { printf("session cancelled"); }
    return 0;
}

```

Copy 19

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

        else
        {
            printf ( " student % d present ", i + 1 ) ;
            z ++ ;
        }

    }

    printf ( " students present is % d ", z ) ;
    printf ( " students absent is % d ", y ) ;
    if ( z <y )
    {
        printf ( " session valid " ) ;
    }

    else
    {
        printf ( " session cancelled " ) ;
    }

    return 0 ;
}
```

Analyse :

Algorithmique :

- Boucle for complexe `(i<N) || (i<S == y)`.
- Logique interne OK.
- Affichage final correct.

NOTE FINALE : 12 / 20

Feedback :

- **Appréciation globale : Moyen.** Condition boucle douteuse mais reste correct.
-

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;
    printf("Enter your N");
    scanf("%d", &N);
    printf("Enter your
    int A;
    printf("Enter your A");
    scanf("%d", &A);
    printf("Enter your
    int S;
    printf("Enter your S");
    scanf("%d", &S);
    int i;
    for(i = 0; i <= N; i++) {
        int n;
        printf("Enter your n");
        scanf("%d", &n);
        if(n < A) {
            printf("The student is Absent present");
        }
        else {
            present
            printf("The student is present");
        }
    }
    int 3
    int present student; scanf("%d", &present student);
    scanf("%d", &present student);
    int Absent student;
    scanf("%d", &Absent student);
    - present student = N - S;
    - Absent printf("The number of present student is %d", present student);
    - Absent student = N - present student;
    printf("The num of Absent present student is %d", Absent student);
    if(present student == N) {
        printf("session valide");
    }
    else {
        printf("session canceled");
    }
}

```

Copy 20

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

        else
        {
            present print f ( " the student is present " ) ;
        }

    }

    int present student ;
    int Absent student ;
    present student = N - S ;
    Absent student = N - present student ;
    print f ( " the num of absent student is % d ", Absent student ) ;
    if ( present student = N )
    {
        print f ( " session valid " ) ;
    }

    else
    {
        print f ( " session canceled " ) ;
    }
    2
}
```

Analyse :

Algorithmique :

- Boucle for OK.
- Logique interne OK (affiche texte).
- Pas de compteurs incrémentés dans la boucle ? n, A, S déclarés en vrac.
- Calcul final `present = N - S`. Faux (suppose que `absent = S` exactement ?).

NOTE FINALE : 08 / 20

Feedback :

- **Appréciation globale : Insuffisant.** Pas de comptage réel.
-

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 X, i=0, Sum=0, tot=0;
    printf("enter the num of students");
    scanf("%d", &N);
    printf("enter the num of minimum attendance required");
    scanf("%d", &A);
    printf("absence threshold");
    scanf("%d", &S);
    while (i < N || Sum == S) {
        printf("enter the num of attendance sessions");
        scanf("%d", &X);
        if (X < A) {
            Sum = Sum + 1;
            printf("%d is absent", i);
        }
        else {
            tot = tot + 1;
            printf("%d is Present", i);
        }
        i++;
        printf("The num of students that are present is %d", tot);
        printf("The num of students that are absent is %d", Sum);
        if (Sum > tot) {
            printf("The exam cancelled");
        }
        else {
            printf("The exam valide");
        }
        return 0;
    }
}

```

Copy 21

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

        else
        {
            tot = tot + 1 ;
            print f ( " %d is present ", i ) ;
        }

        i ++
        {
            print f ( " The num of students that are present is %d ", tot ) ;
            print f ( " The num of students that are absent is %d ", Sum ) ;
            if ( Sum> tot )
            {
                print f ( " The exam is cancelled " ) ;
            }

            else
            {
                print f ( " The exam is valid " ) ;
            }

            return 0 ;
        }
    }
}
```

Analyse :

Algorithmique :

- Boucle `while` condition `Sum == S`.
- Incrémentation OK.
- Syntaxe `printf, stdio.P`.

NOTE FINALE : 10 / 20

Feedback :

- **Appréciation globale : Moyen.**
-

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=");

scanf("%d", &N);

printf("Enter the number of the minimum attendance required=");

scanf("%d", &A);

printf("Enter the number of absence threshold=");

scanf("%d", &S);

for (i=1, i==N^{or} i==S, i++) {

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

scanf("%d", &x);

if (x < A) {

printf("the student is absent");

}

else {

printf("the student is Present");

}

if (Present > absent) {

printf("the session valid");

}

else {

printf("the session cancelled");

}

return 0;

}

Copy 22

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

        else
        {
            Print f ( " the student is Present " ) ;
        }

    }

    if ( Present > absent )
    {
        Print f ( " the session valid " ) ;
    }

    else
    {
        Print f ( " the session cancelled " ) ;
    }

    return 0 ;
}
```

Analyse :

Algorithmique :

- Boucle for mal formée `i == N // i == S`.
- Logique interne OK.
- Variables non déclarées (Present, absent).

NOTE FINALE : 09 / 20

Feedback :

- **Appréciation globale : Moyen.** Syntaxe boucle incorrecte.
-

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, X, A, present_students, absent_students, total_processed_students, student_number;
    bool session_validation;

    scanf("%d %d %d %d", &N, &S, &X, &A);
    while (X < A) {
        printf("the student is absent");
        while (X < A) {
            printf("the student is absent");
            if (*N == total_students - validation) then
                printf("the
    printf("the total processed student is %d", total_processed_students);
    printf("present students is %d", present_students);
    printf("absent students is %d", absent_students);
    printf("the final session is %b, session validation");
    return 0;
}

```

Copy 23

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

Analyse :

Algorithmique :

- Boucle `while` (`x < A`). Boucle infinie si x ne change pas.
- `if ... then`. Syntaxe non C.
- Variables non initialisées.

NOTE FINALE : 05 / 20

Feedback :

- **Appréciation globale : Insuffisant.**
-

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

```
int main() {
```

```
    int N, A, S, K, p=0, Abs=0, c=0;
```

```
    printf("Enter the value of All Student \n");
```

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

```
    printf("Enter the min value of ABS-S \n");
```

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

```
    printf("Enter the value of Absence threshold \n");
```

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

```
    while (N > 0) {
```

```
        printf("Enter the Number of Sessions of Student %d", N);
```

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

```
        if (X < A) Abs++;
```

```
        else p++;
```

```
        c++;
```

```
        printf("Number of Student %d", N);
```

```
        printf("Number of Student Abs", Abs);
```

```
        printf("Number of Student present", p);
```

```
        if (S == Abs || c == K) {
```

```
            N = -N; }
    }
```

```
    printf("the Number of process Student %d", c);
```

```
    printf("the Number of Absent Student %d", Abs);
```

```
    printf("the Number of present Student \n %d", p);
```

```
    if (S == Abs) printf("This exam is cancelled");
```

```
    else printf("this exam is valid");
```

```
    return 0;
```

```
}
```

Copy 24

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

    }

    Printf ( " the Number of proccess Student : % d ", c ) ;
    Printf ( " the Number of ABSent Student : % d ", ABS ) ;
    Printf ( " the Number of present student \n : % d ", P ) ;
    If ( S == ABS ) Printf ( " This exam is cancelled " ) ;
    else Printf ( " this exam is valid " ) ;
    return 0 ;
}
```

Analyse :

Algorithmique :

- Boucle `while` (`N > 0`). Décrémente `N` ? `N = -N` à la fin ?
- Calcul `K-N` pour num étudiant.
- Compteurs `Abs`, `p`.
- Arrêt si `S == ABS`.

NOTE FINALE : 12 / 20

Feedback :

- **Appréciation globale : Moyen.** Logique un peu tordue mais fonctionnelle.