

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

```
int main ( ) {
```

```
int S, N, A, n, L, M.
```

```
printf("Enter the total of Student");
```

```
scanf("%d", &N); while (n < N)
```

```
printf("Enter n");
```

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

```
n = 0
```

```
if (n < A) {
```

```
printf("the Student is absent");
```

```
else printf("the Student is present");
```

```
End if
```

```
while
```

```
M = N + 1
```

```
if (M
```

```
L = N + 1
```

```
if (M < A)
```

```
printf("the number of Student present");
```

```
else printf("the number of Student absent");
```

```
S = N + 1
```

```
if (L < S)
```

```
End while
```

```
printf("the session valid");
```

Copy number :

1-BIS

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

```
else (
    printf("Session cancelled")
End while
} roller 0 }
```

Copy 1

```
#include <stdio.h>
int main()
{
    int S, N, A, n, L, m;
    printf("Enter the total of Student");
    scanf("%d", &N);
    while (N> 44)
    {
        printf("Enter n");
        scanf("%d", &n);
        n = 0;
        if (n <A)
        {
            printf("the Student is abzent");
        }

        else
        {
            printf("the Student in present");
        }

        int i = 0;
        m = n + i;
        L = n + i;
        if (m <A)
        {
            printf("the number of Student present");
        }

        else
        {
            printf("the number of Student abzent");
        }

        S = n;
        i++;
        if (L <S)
        {
            // End while
        }

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

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

```
    }  
}  
return 0;  
}
```


Analyse :

Algorithmique :

- Boucle `while (N > 44)`. Condition absurde et "magique".
- `n` est utilisé pour la lecture de `N` ?? Non `scanf ("%d", &N)`. `n` lu dans la boucle mais écrasé par `n = 0`.
- Confus, utilisation de constantes magiques.
- Termes incorrects (abzent).

NOTE FINALE : 03 / 20

Feedback :

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

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

#include <stdio.h>

int main() {

int N, A, s, ac, sA, sP, sT;

printf("Enter the total number of registered student: ");

scanf("%d", &N);

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

scanf("%d", &A);

printf("Enter the absence threshold: ");

scanf("%d", &s);

~~printf~~ for(int i=0; i<=N; i++){

printf("Enter x: ", i+1);

scanf("%d", &x);

if(x < A){

printf("the student is absent!");

} else {

printf("the student is present!");

}

sP = N - x;

sA = N - sP;

printf("the student present students = %d", sP);

printf("the absent students = %d", sA);

if(x == N || x == s){

printf("simulation stops");

}

}

(1)

```
ST = SP + SA;
```

```
printf("the total processed students = %d", ST);
```

```
printf("the present students = %d", SP);
```

```
printf("the absent students = %d", SA);
```

```
// Final status:
```

```
if (N < SA) {
```

```
    printf("session cancelled!");
```

```
} else {
```

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

```
}
```

```
return 0;
```

```
}
```

(2)

Copy 2

```
#include <stdio.h>
int main()
{
    int N, A, S, x, sA, sP, sT;
    printf("Enter the total number of registered student : ");
    scanf("%d", &N);
    printf("Enter the minimum attendance required : ");
    scanf("%d", &A);
    printf("Enter the absents threshold : ");
    scanf("%d", &S);
    for (int i = 0; i <= N; i++)
    {
        printf("Enter x : ", i + 1);
        scanf("%d", &x);
        if (x < A)
        {
            printf("the student is absent.");
        }

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

        sP = N - x;
        sA = N - sP;
        printf("the student present students = %d", sP);
        printf("the absent students = %d", sA);
        if (x == N || x == S)
        {
            printf("simulation stops");
            sT = sP + sA;
            printf(" the total processed students = %d", sT);
            printf(" the present students = %d", sP);
            printf(" the absent students = %d", sA);
            // Final status: if (N < sA)
            {
                printf(" session cancelled!");
            }

            else
            {
                printf(" session valid");
            }

            return 0;
        }
    }
}
```

}

Analyse :

Algorithmique :

- Lecture OK.
- Boucle for.
- Calculs finaux dans la boucle : $SP = N - x$?? x est le nombre de séances de l'étudiant, pas le nombre de présents ! Grosse confusion sur la signification de x .
- Condition d'arrêt `if (x == N || x == S)`. Encore confusion entre x (séances d'un itudiant) et les compteurs globaux.
- Ne compte rien (variables écrasées à chaque tour).

NOTE FINALE : 06 / 20

Feedback :

- **Appréciation globale : Insuffisant.** Confusion totale sur les variables.
-

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; int Absence, Presence;
    printf("what is the total number of registered Students : ");
    scanf("%d", &N);
    printf("what is the minimum attendance required : ");
    scanf("%d", &A);
    printf("what is the absence threshold:");
    scanf("%d", &S);
    for (i = 1; i <= N; i++) {
        printf("Student number %d: ", i);
        scanf("%d", &x);
        if (x < A) {
            Absence = Absence + 1;
        } else {
            Presence = Presence + 1;
        }
        printf("Present Students: %d\n", Presence);
        printf("Absent Students: %d", Absence);
        if (i == N || Absence == S) {
            if (Presence > Absence) {
                printf("Total processed Students: %d", i);
                printf("Present Students: %d", Presence);
                printf("Absent Students: %d", Absence);
                printf("Session Valid");
            } else {
                return 0;
            }
        } else {
            printf("Total processed Students: %d", i);
            printf("Present Students %d", Presence);

```


Copy number :

3-BIS

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

```
Printf("Absent Students: %d = Absence");  
Printf("Session Cancelled! = ");  
Return 0; }  
}
```

```
}  
}  
// I'm Sorry for my larg / bad hand writing :c
```

Copy 3

```
#include <stdio.h>
int main()
{
    int N, A, S;
    int x;
    int i;
    int Absence = 0, Presence = 0;
    printf("what is the total number of registered Students : ");
    scanf("%d", &N);
    printf("what is the minimum attendance required : ");
    scanf("%d", &A);
    printf("what is the absence threshold : ");
    scanf("%d", &S);
    for (i = 1; i <= N; i++)
    {
        printf("Student number %d", i);
        scanf("%d", &x);
        if (x < A)
        {
            Absence = Absence + 1;
        }

        else
        {
            Presence = Presence + 1;
        }

        printf("Present Students : %d\n", Presence);
        printf("Absent Students : %d", Absence);
        if (i == N || Absence == S)
        {
            if (Presence > Absence)
            {
                printf("Total processed Students : %d", i);
                printf("Present Students : %d", Presence);
                printf("Absent Students : %d", Absence);
                printf("Session Valid");
                break;
            }

            else
            {
                printf("Total processed Students : %d", i);
                printf("Present Students %d", Presence);
                printf("Absent Students : %d", Absence);
                printf(" Session Cancelled ! ");
                return 0;
            }
        }
    }
}
```

```
}
```

```
}
```

```
}
```

Analyse :

Algorithmique :

- Lecture OK.
- Boucle for. Logique interne OK.
- Compteurs Absence, Presence.
- Arrêt : `if (i == N || Absence == S)`. Utilise break (interdit) ou return.
- Affiche status final DANS la boucle (à chaque itération si condition remplie).

NOTE FINALE : 14 / 20

Feedback :

- **Appréciation globale : Moyen / Bon.** Pénalité pour break et structure "tout dans la boucle".
-

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

```

#include <stdio.h>
#include <math.h>
int main() {
    int N;
    int A;
    int S;
    printf("enter minimum attendance required");
    scanf("%d", &A);
    printf("enter the absence threshold");
    scanf("%d", &S);
    for (i = 1; i <= N; i++) {
        printf("enter the total number of students: \n");
        scanf("%d", &N);
        printf printf("Students number %d: \n", i);
        i = i + 1;
        int x;
        scanf("%d", &x);
        printf("Student number %d : %d", i, x);
        if (x < A) {
            printf("the student is absent: \n");
        } else {
            printf("the student is present: \n");
        }
    }
}

```

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

```

}
int 4 (طريق) ;
int present;
int = N - present;

printf ("the absent is %d", int);
int sum;
sum = absent

if (present < 5) {
    printf ("Stop program");
} else if (N = N) {
    printf ("Stop program");
}

return 0;
}

```

Copy 4

```
#include <stdio.h>
#include <math.h>
int main()
{
    int N;
    int A;
    int S;
    int i;
    printf("enter minimum attendance required");
    scanf("%d", &A);
    printf("enter the absence threshold");
    scanf("%d", &S);
    for (i = 1; i <= N; i++)
    {
        printf("enter the total number of students : \n");
        scanf("%d", &N);
        printf(". Students number %d : \n", i);
        i = i + 1;
        int x;
        scanf("%d", &x);
        printf(" Student number %d : %d", i, x);
        if (x < A)
        {
            printf("the Student is absent : \n");
        }

        else
        {
            printf("the Student is present \n");
        }

    }

    int present;
    int absent = N - present;
    printf(" the absent is %d", absent);
    int Sum;
    Sum = absent;
    if (present < S)
    {
        printf(" Stop program");
    }

    else if (N == N)
    {
        printf(" Stop program");
    }

    return 0;
```


}

Analyse :

Algorithmique :

- Pas de lecture initiale de N ! (Variable non init).
- Boucle for (i=1; i<=N). N aléatoire.
- Lecture de N **DANS** la boucle !?
- Modification de i (i = i + 1) dans le for (saute des étapes).
- absent = N - present. present non init.

NOTE FINALE : 04 / 20

Feedback :

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

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

```
int main () {
    int processed = 0;
    int N = 0;
    int A = 0;
    int s = 0;
    int X = 0;
    int absent = 0;
    int present = 0;
```

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

```
printf("Enter the min attendance
Required: ");
```

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

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

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

```
printf("student number.
%d\n", i);
```

```
printf("%d absent
students\n", absent);
```

```
printf("%d present
student s\n", present);
```

```
if (absent >= s) {
    i = N;
}
```

Copy number : 5-BIS

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

```
printf("total number of pro  
processed = absent + present ;  
printf("the total number of processed students  
is: %d", processed);  
printf("%d present students", present);  
printf("%d absent students", absent);  
  
if (absent >= 8)  
printf("valid session");  
else  
printf("cancelled session");  
  
return 0;  
}
```

Copy 5

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

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

        printf("student number : %d\n", i);
        printf("%d absent students\n", absent);
        printf("%d present students\n", present);
        if (absent >= S)
        {
            i = N;
        }
    }

    int processed = absent + present;
    printf("the total number of processed students is : %d", processed);
    printf("%d present students", present);
    printf("%d absent students", absent);
    if (absent >= S)
    {
        printf("valid Session");
    }
}
```

```
    }  
  
    else  
    {  
        printf("Cancelled Session");  
    }  
  
    return 0;  
}
```

Analyse :

Algorithmique :

- Lectures OK.
- Boucle for.
- Logique interne OK.
- Arrêt : `if (absent >= S) i = N;`. (Hack de sortie, ok).
- Condition finale inversée : `if (absent >= S) -> Valid ??` Non, cancelled ! (Logique `valid` si `absent >= S` est faux).

NOTE FINALE : 14 / 20

Feedback :

- **Appréciation globale : Moyen.** Attention à la condition finale (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 N, X, A, S, L = 0, M = 0, i = 1;

printf("type total number of Students);

scanf("%d", &N);

printf("type the minimum of attendance required);

scanf("%d", &A);

printf("type the absence threshold);

scanf("%d", &S);

while(L > S && i <= N) {

printf("Student %d:", i);

scanf("%d", &X);

if(X < A) {

L++; }

else {

M++; }

printf("number of presents %d:", M);

printf("number of absents %d:", L); }


```
if (L==S) {  
    printf("The exame is cancelled");  
}  
else {  
    printf("The exame is valid");  
    printf("number of Student total %d", N);  
    printf("number of Student absent %d", L);  
    printf("number of Student present %d", M);  
    return 0;  
}
```

Copy 6

```
#include <stdio.h>
int main()
{
    int N, X, A, S, L = 0, M = 0, i = 0;
    printf("type total number of Students");
    scanf("%d", &N);
    printf("type the minimum of attendance required");
    scanf("%d", &A);
    printf("type the absence threshold");
    scanf("%d", &S);
    while ((L) < S && i < N)
    {
        printf("Student %d", i);
        scanf("%d", &X);
        if (X < A)
        {
            L++;
        }

        else
        {
            M++;
        }

        i++;
    }

    if (L == S)
    {
        printf(" the exame is cancelled ");
    }

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

    printf(" number of Student total : %d ", N);
    printf(" number of Student absent : %d ", L);
    printf(" number of Student present : %d ", M);
    return 0;
}
```

Analyse :

Algorithmique :

- Boucle while `((L) < S && i < N)`. Condition composée parfaite. L = absents.
- Logique interne simple et correcte.
- Affichage final correct.

NOTE FINALE : 18 / 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 n, A, S, S1 = 0, n1 = 0, k;
    printf("enter total number of registered students \n");
    scanf("%d", &n);
    printf("enter minimum attendance required \n");
    scanf("%d", &A);
    printf("enter absense threshold");
    scanf("%d", &S);
    for(int i = 0; i <= n; i++) {
        if(S1 <= S) {
            printf("the numbre of attended sessions X: \n");
            scanf("%d", &se);
            if(se < A) {
                S1 = S1 + 1;
                printf("student numbre: %d - Present students: %d\n", i, n1, S1);
                printf("absent student: %d \n", i, n1, S1);
            } else {
                n1 = n1 + 1;
                printf("student numbre: %d - Present students\n", i, n1, S1);
                printf("absent student: %d \n", i, n1, S1);
            }
        } else {
            n = i;
            printf("absent student: %d \n", i, n1, S1);
        }
    }
}

```

Copy number :

7-BIS

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

```
if (s1 == s && n1 >= n) {  
    printf("Session valid \n");  
    cancelled  
}  
else {  
    printf("Session valid \n");  
}  
printf("\n");  
return 0;  
}
```

Copy 7

```
#include <stdio.h>
int main()
{
    int n, A, S, S1 = 0, n1 = 0, x;
    printf("enter total nombre of registered students \n");
    scanf("%d", &n);
    printf("enter minimum attendance required \n");
    scanf("%d", &A);
    printf("enter absense threshold");
    scanf("%d", &S);
    for (int i = 0; i <= n; i++)
    {
        if (S1 <= S)
        {
            printf("the nombre of attended sessions x : \n");
            scanf("%d", &x);
            if (x < A)
            {
                S1 = S1 + 1;
                printf("student nombre: %d - Present students: %d - absent student: %d \n",
                    i, S1, n - S1);
            }
            else
            {
                n1 = n1 + 1;
                printf("student nombre: %d - Present students : %d - absent student: %d \n",
                    i, S1, n - S1);
            }
        }
        else
        {
            n = i;
        }
    }

    if (S1 == S || n1 >= n)
    {
        printf("Session cancelled \n");
    }
    else
    {
        printf("Session valid \n");
    }

    printf("\n");
    return 0;
}
```

}

Analyse :

Algorithmique :

- Boucle for. $i \leq n$ (utilise n lu pour nombre étudiants).
- Condition arrêt $\text{if } (S1 \leq S) \text{ (Absents} \leq \text{Seuil)}$.
- Sinon else $\{ n = i; \}$. Hack de sortie (met fin à la boucle for car i va dépasser n ? Non, $n=i$ rend $i \leq n$ vrai tant que i n'augmente pas. $i++$ fera $i > n$ au tour suivant. Astucieux mais n est écrasé, donc on perd le nombre total.
- Pas critique car i contient le nombre traité.

NOTE FINALE : 15 / 20

Feedback :

- **Appréciation globale : Bon.**

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

```

#include <stdio.h>
int main() {
    int N, A, S, NA = 0, NP = 0, X;
    printf("enter the number of N and A and S");
    scanf("%d%d%d", &N, &A, &S);
    for (i = 1; i <= N; i++) {
        printf("enter the number of attended Session %d", i);
        scanf("%d", &X);
        if ("X < A") {
            NA = NA + 1;
        }
        else {
            NP = NP + 1;
        }
        printf("The number of present Students: %d", NP);
        printf("The number of Absent Students: %d", NA);
        if (NA > S) {
            printf("Session cancelled");
        }
    }
}

```

```
printf ("the number of present Students %d", NP);  
printf ("the number of Absent Students %d", NA);  
  
if (NP < 5) {  
    printf ("Session valid");  
}  
else {  
    printf ("Session cancelled");  
}  
  
set  
return 0;  
}
```

Copy 8

```
#include <stdio.h>
int main()
{
    int N, A, S, NA = 0, NP = 0, x, i;
    printf("entre the number de N and A and S");
    scanf("%d %d %d", &N, &A, &S);
    for (i = 1; i <= N; i++)
    {
        printf("entre the number of attended Sessin x : %d", i);
        scanf("%d", &x);
        if (x < A)
        {
            NA = NA + 1;
        }

        else
        {
            NP = NP + 1;
        }

        printf("the number of present students : %d", NP);
        printf("the number of Absent students : %d", NA);
        if (NA >= S)
        {
            printf("Session cancellad");
        }

    }

    printf("the number of present students %d", NP);
    printf("the number of Absent students %d", NA);
    if (NP < S)
    {
        printf("Session valid");
    }

    else
    {
        printf("Session cancellad");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Lecture OK.
- Boucle for.
- Logique interne correcte (Compteurs NA, NP).
- Condition finale `if (NP < S)`. Erreur: c'est le seuil d'ABSENCE (NA) qui compte, pas la présence. Donc valid si `NA < S` (et non NP). Là il dit valid si `NP < S` (peu de présents -> valide ? Contresens).

NOTE FINALE : 14 / 20

Feedback :

- **Appréciation globale : Moyen.** Confusion sur la condition de validité finale.
-

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;

    printf("Entre the number of attended sessions X");
    scanf("%d", &X);
    if (X < A) {
        printf("the student is absent");
    }
    else
        printf("the student is present");
    while (i < N) {
        absent = S
        while (i < N . absent < S) {
            printf("The number of attended sessions X");
            scanf("%d", X)
            if (X < A) {
                printf("the student is absent");
            }
            else
                printf("the student is present");
            i = i + 1
        }
    }
}

```


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, R=0, M=0, i=1;
    printf("enter total number of registered students:");
    scanf("%d", &N);
    printf("أدخل العدد الأدنى للدخول المطلوب :");
    scanf("%d", &A);
    printf("أدخل العتبة الغيابات المسموح بها :");
    scanf("%d", &S);
    for(i=1; i<=N; i++) {
        printf("أدخل عدد المصنف الذي دفعه الطالب :");
        scanf("%d", &X);
        if(X < A) {
            R = R + 1; // إذا كان طالب غائب يضيف 1 إلى عدد الطلبة غائبين
        }
        else {
            M = M + 1; // إذا كان طالب حاضراً يضيف 1 إلى عدد الطلبة حاضرين
        }
        printf("%d و رقم الطالب هو : %d", i, X);
        printf("M و عدد الطلبة الحاضرين : %d", M);
        printf("R و عدد الطلبة غائبين : %d", R);
    }
    printf("M و عدد الطلبة الحاضرين : %d", M);
    printf("R و عدد الطلبة الغائبين : %d", R);
    if (M=N || R<S) {
        printf("الإمتحان صالح");
    }
    else {
        printf("الإمتحان ملغى");
    }
    return 0;
}

```


Copy 9

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

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

    while (i < N && absent < S)
    {
        printf("The number of attended sessions x");
        scanf("%d", &x);
        if (x < A)
        {
            printf("the student is absent");
        }

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

        i = i + 1;
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Lecture de x avant la boucle.
- Premier test hors boucle.
- Boucle while (i < N && absent < S). i et absent non initialisés.
- Lecture x dans la boucle. Pas de compteurs incrémentés dans la boucle (juste des prints).
- Boucle ne s'arrêtera jamais sur absent (car non incrémenté).

NOTE FINALE : 07 / 20

Feedback :

- **Appréciation globale : Insuffisant.** Variables non initialisées et compteurs manquants.
-

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("total number of registered student");
    scanf("%d", &N);
    printf("the minimum attendance req required");
    scanf("%d", &A);
    printf("absent threshold");
    scanf("%d", &S);
    printf("is you present write 1");
    scanf scanf("%d", &X);
```

~~(X < A) {~~
~~printf("you~~

```
for (i=0; i <= N; i++) {
    i = x + i;
    if (x < A) {
        printf("the student is absent");
    }
    else {
        scanf("%d", &X);
    }
}
```

```
if (x == N 4 || x == S) {
```

```
    printf("the exam ended") }
```

```
else
```

```
    printf("total processed:%d", N);
```

```
    int Z;
```

```
    Z = N - x;
```

```
    printf("absent student (%d) %d", Z);
```

Copy 10

```
#include <stdio.h>
int main()
{
    int N, A, S, x;
    printf("total number of registred student");
    scanf("%d", &N);
    printf("the minimum attendance req required");
    scanf("%d", &A);
    printf("absens thoreshold");
    scanf("%d", &S);
    printf("is you present write 1");
    scanf("%d", &x);
    for (int i = 0; i <N; i++)
    {
        if (x <A)
        {
            printf("the student is apsent");
        }

        else
        {
            scanf("%d", &x);
        }

    }

    if (x == N || x == S)
    {
        printf("the exam ended");
    }

    else
    {
        printf("total processed %d", N);
    }

    int z;
    z = N - x;
    printf("absent studet : %d", z);
    return 0;
}
```

Analyse :

Algorithmique :

- `printf("is you present write 1")`. Ne suit pas l'énoncé (il faut lire le nombre de séances).
- Boucle `for`.
- Si `x < A` -> Absent. Sinon `scanf("%d", &x)`. (Lit une nouvelle valeur si présent ?).
- Pas de compteurs.
- Condition finale `if (x == N)`.

NOTE FINALE : 04 / 20

Feedback :

- **Appréciation globale : Très Insuffisant.** Ne respecte pas l'énoncé.
-

```

#include <stdio.h>
int main()
{
    int N, S, A;
    int x;
    int present-students = 0;
    int absent-students = 0;
    int total-processed = 0;
    printf("total processed student");
    scanf("%d", &N);
    printf("minimum attendance required");
    scanf("%d", &A);
    printf("absence threshold");
    scanf("%d", &S);
    while (current-step < N && absent-students < S)
    {
        current-step = current-students + 1;
        printf("/n current-step %d \n", current-step);
        scanf("%d", &x);
        if (x < 1)
        {
            absent-students = absent-students + 1;
        }
        else
        {
            present-students = present-student + 1;
        }
        printf("%d\n", current-step);
        printf("%d\n", present-students);
        printf("%d\n", absent-students);
    }
}

```

1/2

Copy number : 11-1315

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

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

```
printf("total processed students : %d\n", current - step);
printf("Final present - students : %d\n", presents - students);
printf("Final absent - students : %d\n", absents - students);
if (absent - students >= 5) {
    printf("Session cancelled\n");
} else {
    printf("Session valid\n");
} else
    return 0;
}
```

2/2.

Copy 11

```
#include <stdio.h>
int main()
{
    int N, S, A;
    int x;
    int present_students = 0;
    int absent_students = 0;
    int total_processed = 0;
    int current_step = 0;
    printf("total processed student");
    scanf("%d", &N);
    printf("minimum attendance required");
    scanf("%d", &A);
    printf("absence threshold");
    scanf("%d", &S);
    while (current_step < N && absent_students < S)
    {
        current_step = current_step + 1;
        printf("\n current - step %d \n", current_step);
        scanf("%d", &x);
        if (x < 1)
        {
            absent_students = absent_students + 1;
        }

        else
        {
            present_students = present_students + 1;
        }

        printf("%d \n", current_step);
        printf("%d \n", present_students);
        printf("%d \n", absent_students);
    }

    printf("total processed students : %d \n", current_step);
    printf("Final present - students : %d \n", present_students);
    printf("Final absents - students : %d \n", absent_students);
    if (absent_students >= S)
    {
        printf("Session cancelled \n");
    }

    else
    {
        printf("Session valid \n");
    }

    return 0;
}
```

}

Analyse :

Algorithmique :

- Boucle `while` avec condition composée correcte.
- Logique interne : `if (x < 1)`. Pourquoi 1 ? L'énoncé dit `if (x < A)`. L'étudiant utilise une constante magique.
- Sinon logique correcte.

NOTE FINALE : 13 / 20

Feedback :

- **Appréciation globale : Moyen.** Utilisation de constante magique 1 au lieu de A.
-

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;
    int x;
    int o = عدد الطلبة الحاضرين ;
    int o = عدد الطلبة الغائبين ;
    int o = إجمالي الطلبة الذين أخذوا معاهذهم ;
    printf("Enter total number of student (N)");
    scanf("%d", &N);
    printf("Enter A");
    scanf("%d", &A);
    printf("Enter S");
    scanf("%d", &S);

    while (total processed < S) {
        printf("Enter the number of attended session %d", x);
        scanf("%d", &x);
        if (x < A) {
            absent = absent + 1;
        } else {
            present = present + 1;
            printf("student %d : present", x);
            printf("student %d : absent", x);
            x = x + 1;
            N = N - 1;
        }
    }
}

```

Copy number : 12-BIS

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

```
printf("total student processed: %d\n", i - 1);  
printf("present student = %d\n", present);  
printf("absent student = %d\n", absent);  
  
if (A <= 5) {  
    printf("session valid");  
} else {  
    printf("session cancelled");  
}  
  
return 0;
```

Copy 12

```
#include <stdio.h>
int main()
{
    int A, N, S;
    int x;
    int present = 0;
    int absent = 0;
    int total_processed = 0;
    int i = 0;
    printf("entre total number of student (N)");
    scanf("%d", &N);
    printf("Enter A");
    scanf("%d", &A);
    printf("Enter S");
    scanf("%d", &S);
    while (total_processed != N && absent < S)
    {
        printf("Enter the number of attended session x :");
        scanf("%d", &x);
        if (x < A)
        {
            absent = absent + 1;
            printf("student %d : absent", i);
        }

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

        i = i + 1;
        // N = N - 1;
        // Logic adjusted based on context total_processed++;
    }

    printf("total stuedent procesed : %d", i);
    printf("presnt student = %d \n", present);
    printf("Absent = %d", absent);
    if (absent <= S) // Adjusted logic based on standard problem statement (usually absent
    {
        printf("ssession valiad");
    }

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



```
    return 0;  
}
```

Analyse :

Algorithmique :

- Boucle while (total != N && absent < S). Correcte.
- Logique interne correcte.
- Incrémentation des compteurs OK.
- Affichage final OK.

NOTE FINALE : 18 / 20

Feedback :

- **Appréciation globale : Très Bon.**

```

#include <stdio.h>
int main() {
    int n, x, A, s, Present = 0, absent = 0, session;
    printf("entre number of students");
    scanf("%d", &n);
    printf("entre minimum attendance");
    scanf("%d", &A);
    printf("entre absence threshold");
    scanf("%d", &s);
    for (i = 1; i <= n; i++) {
        printf("entre number of attendance for the student  
number %d %d", i, x);
        if (x < A) {
            printf("the student number is absent");
            absent = absent + 1;
        }
        else {
            printf("the student is present");
            Present = Present + 1;
        }
        if (absent > s) {
            printf("max absence reached");
            return 1;
        }
    }
}

```

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

```

If (present > absent) {

```

```

    session = 1;

```

```

}

```

```

Else {

```

```

    session = 0;

```

```

}

```

```

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

```

```

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

```

```

If (session) {

```

```

    printf("session valid");

```

```

Else {

```

```

    printf("session cancelled");

```

```

}

```

```

return 0;

```

```

}

```

Copy 13

```
#include <stdio.h>
int main()
{
    int n, x, A, S, present = 0, absent = 0, session;
    printf("entre number of students");
    scanf("%d", &n);
    printf("enter minimum attendance :");
    scanf("%d", &A);
    printf("entre absence thershold");
    scanf("%d", &S);
    for (int i = 1; i <= n; i++)
    {
        printf("entre number of attendance for the student number %d : %d ", i, x);
        if (x < A)
        {
            printf("the student numbe is absent");
            absent = absent + 1;
        }

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

        if (absent > S)
        {
            printf("max absence reched");
            return 1;
        }

    }

    if (present > absent)
    {
        session = 1;
    }

    else
    {
        session = 0;
    }

    printf("the number of present students : %d ", present);
    printf("the number of absent students : %d ", absent);
    if (session)
    {
        printf("session valid");
    }
}
```

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

return 0;
}
```

Analyse :

Algorithmique :

- Lecture OK.
- Boucle for.
- x n'est pas lu DANS la boucle (utilisé dans printf ?). Le scanf de x est absent ? Non, il y a scanf ("%d", &x)... ah non, scanf n'est pas visible dans l'extrait fourni dans code_doc ! S'il est absent du code, c'est grave. (Atten, ligne 21 fait printf(. . . , x) mais x n'est pas lu avant).
- Si absent > S, arrêt prématuré (return 1).
- Logique finale (session) correcte bien que verbeuse.

NOTE FINALE : 09 / 20

Feedback :

- **Appréciation globale : Moyen.** Oubli probable de la lecture 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, S;
    int i, X, ab = 0, p = 0;
    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 \n");
    scanf("%d", &S);
    for (i = 1; i <= N; i++)
    {
        printf("Enter the number of attended session for student: %d \n", i);
        scanf("%d", &X);
        if (X < A)
        {
            printf("the student is absent");
            ab = ab + 1;
        }
        else
        {
            printf("the student is present");
            p = p + 1;
        }
    }
    printf("the number of present student is: %d \n", p);
    printf("the number of absent students is: %d \n", ab);
}

```

Copy number : ~~14-1315~~ 14-1315

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

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

```
if (i != N || a < 8)
{
    printf("the total number of present student is: %d \n", p);
    printf("the total number of absent students is: %d \n", ab);
    if (ab > p)
    {
        printf("session cancelled");
    }
    else
    {
        printf("Session valid");
    }
    else
    {
        printf("Simulation was stopped");
    }
    return 0;
}
```

Copy 14

```
#include <stdio.h>
int main()
{
    int N, A, S;
    int i, x, ab = 0, p = 0;
    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 \n");
    scanf("%d", &S);
    for (i = 1; i <= N; i++)
    {
        printf("Enter the number of attended session for student : %d \n", i);
        scanf("%d", &x);
        if (x < A)
        {
            printf("the student is absent");
            ab = ab + 1;
        }

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

        printf("the number of present student is : %d \n", p);
        printf("the number of absent students is : %d \n", ab);
    }

    if (i == N || ab < S)
    {
        printf("the total number of present student is : %d \n", p);
        printf("the total number of absent students is : %d \n", ab);
        if (ab > p)
        {
            printf("session cancelled");
        }

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

    else
    {

```

```
        printf("Simulation was stoped");  
    }  
  
    return 0;  
}
```

Analyse :

Algorithmique :

- Lecture OK.
- Boucle for. Logique interne OK.
- Condition structurelle bizarre après la boucle : `if (i = !N || ab < S)`. Affectation `i = !N` (vaut 0 ou 1) !
Erreur syntaxe majeure.
- Comparaison finale `absent > present` pour annuler ? Critère pas dans l'énoncé (c'est `> S`).

NOTE FINALE : 12 / 20

Feedback :

- **Appréciation globale : Moyen.** Erreur fatale dans le `if` final (=).
-

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

#include <stdio.h>

int S, A, N;

int x;

int o = 0, k = 0;

printf("ادخل عدد الطلبة N ");

scanf("%d", &N);

printf("ادخل الاموال التي للحضور ");

scanf("%d", &A);

printf("ادخل نسبة الغياب S ");

scanf("%d", &S);

while (S < 5) {

printf("الغالبه + 1 - الطالب رفع الاموال\n");

scanf("%d", &x);

if (x < A) {

k++;

}

else {

o++;

S++;

printf("معالجة : \n", o);

printf("حاضر : \n", k);

printf("غائب : \n", S);

Copy number : 15-BIS

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

```
{ (s = > 3) if
    printf("هذه الحالة النهائية الامتحان ملغى");
} else {
    printf("هذه الحالة النهائية الامتحان صالح");
}
return 0;
}
```



```
#include <stdio.h>
int main()
{
    int S, A, N;
    int x;
    int absent = 0, present = 0;
    printf("Enter number of students N");
    scanf("%d", &N);
    printf("Enter minimum attendance A");
    scanf("%d", &A);
    printf("Enter absence threshold S");
    scanf("%d", &S);
    while (N <= S && S > absent)
    {
        printf("Student number %d enter attended sessions", i);
        scanf("%d", &x);
        if (x < A)
        {
            absent++;
        }

        else
        {
            present++;
        }

        i++;
        printf("Processing %d : Present %d / Absent %d \n", i, present, absent);
        if (S <= absent)
        {
            printf("Final Status: Exam Cancelled");
        }

        else
        {
            printf("Final Status: Exam Valid");
        }

    }

    return 0;
}
```

Analyse :

Algorithmique :

- Condition `while (N < i && S > absent)`. `i` non initialisé (donc poubelle). Si `i` hasard > N, boucle s'exécute pas.
- Incrémentation `i++` à la fin.
- Logique interne correcte.

NOTE FINALE : 06 / 20

Feedback :

- **Appréciation globale : Insuffisant.** Variable boucle non initialisée.
-

```

#include <stdio.h>

int main () {
    int N, A, S, X, i;
    printf("enter total number of registered students, N: ");
    scanf("%d", &N);

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

    for (i = 1; i <= N; i++) {
        for (i = 0; i <= S; i++) {
            printf("enter the number of attended sessions, X");
            scanf("%d", &X);

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

            if (X < A) {
                printf("the students is absent");
            }
            else {
                printf("the students is present");
            }
        }
    }
}

```

Copy number : 16-BIS

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

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

```
printf( total processed students );
```

```
if( the students is presents ) {
```

```
printf( session valid );
```

```
else
```

```
printf( session cancelled );
```

```
}
```

```
return 0;
```

```
}
```

Copy 16

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

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

        }

    }

    printf("total processed students");
    if (present > absent)
    {
        printf("Session valid");
    }

    else
    {
        printf("session cancelled");
    }

    return 0;
}
```

Analyse :

Algorithmique :

- Boucles imbriquées `for (i=1;...N) { for (i=0;...S) }`. Réutilise `i` pour la boucle interne ! Casse la boucle externe.
- Lit A à chaque tour ?
- `present`, `absent` non déclarés.

NOTE FINALE : 03 / 20

Feedback :

- **Appréciation globale : Très Insuffisant.**