

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

• The correct of the exercise : • examination attendance monitoring

```

#include <stdio.h>
int main () {
    int n, a, s;
    printf ("enter number :");
    for (j = 0, j <= n, j++); {
        printf ("read the number of attended session n");
        if (n < A)
            printf ("The student is considered absent");
        else
            printf ("The student is present");
        j
        if (n == 0)
            printf ("all N student are processed or the number of absent
            student reaches");
        while ("N >= n")
            present student = N - S
            absent student = N - A
            if (S > 5)
                printf ("session cancelled")
            else {
                printf ("session valid");
                scanf ("total number of student present and absent : %d\n");
            }
            return 0;
    }
}

```

Copy 15

```
. The correct of the exercise : - exomination attendance monitoring #include <stdio.h>
int main ( )
{
    int n, a, S ;
    printf ( " enter number : " ) ;
    for ( j = 0 ; j <= x, j ++ ) ;
    {
        printf ( " read the number of attented session x " ) ;
        if ( x < A ) printf ( " The student is considered absent " ) ;
        else printf ( " The student is present " ) ;
    }

    if ( x == 0 ) printf ( " all N student are processed or the number of absent student re
    while ( " N>= x " ) present student = N - S absent student = N - A if ( S > 5 ) printf (
    {
        printf ( " session valid " ).
    }

    Scanf ( " total number of student present and absent : % d \n ) ;
    return 0 ;
}
```

Analyse :

Algorithmique :

- Boucle `for` (`j=0 ; j<=x . . .`). `x` non lu.
- Mélange instructions et texte.
- Utilisation de chaînes comme conditions `while` (`"N>=x"`).

NOTE FINALE : 01 / 20

Feedback :

- **Appréciation globale : Incompilable.**
-