

La funzione “**dividi**”

Il codice tenta di eseguire una divisione intera tra due numeri (a e b), ma utilizza l'operatore modulo % invece che l'operatore di divisione /. L'operatore % restituisce il resto della divisione intera, non il risultato della divisione stessa. Per eseguire la divisione correttamente, dovrebbe utilizzare l'operatore /.

La correzione per la funzione **dividi()**:

```
void dividi() {
    int a, b;
    printf("Inserisci il numeratore: ");
    scanf("%d", &a);
    printf("Inserisci il denominatore diverso da zero: ");
    scanf("%d", &b);

    if (b != 0) {
        int divisione = a / b;
        printf("La divisione tra %d e %d è: %d\n", a, b, divisione);
    } else {
        printf("Impossibile dividere per zero!\n");
    }
}
```

La funzione **ins_string()**:

La funzione ‘scanf("%s", &stringa)’; legge una stringa di caratteri dalla console, ma non specifica il limite massimo di caratteri da leggere, il che può causare un overflow del buffer se l'utente inserisce più di 10 caratteri. È preferibile utilizzare ‘fgets()’ per leggere una stringa specificando la dimensione massima del buffer per evitare questo problema.

La correzione per la funzione ‘ins_string()’ utilizzando ‘fgets()’:

```
void ins_string() {
    char stringa[10];
    printf("Inserisci la stringa: ");
    fgets(stringa, sizeof(stringa), stdin);
}
```