

Il programma è un assistente digitale che mira ad aiutare l'utente ad eseguire semplici operazioni: moltiplicazione, divisione e l'inserimento di una stringa.

```
int main()
{
    char scelta = {'\0'};
    menu();
    scanf ("%d", &scelta);

    switch (scelta)
    {
```

Errore n°1^[1] : La notazione per la scansione di un carattere non è corretta, "%d" viene utilizzata per un numero intero, mentre in questo caso la risposta attesa è un carattere.

```
switch (scelta)
{
    case 'A':
        moltiplica();
        break;
    case 'B':
        dividi();
        break;
    case 'C':
        ins_string();
        break;
}
```

Errore n°2^[2]: Il programma non prende in considerazione la possibilità che l'utente possa inserire una scelta diversa da A, B e C.

```
void moltiplica ()
{
    short int a,b = 0;
    printf("Inserisci i due numeri da moltiplicare:");
    scanf ("%f", &a);
    scanf ("%d", &b);

    short int prodotto = a * b;

    printf ("Il prodotto tra %d e %d è: %d", a,b,prodotto);
}
```

Errore n°3A^[3]: Nella funzione "moltiplica" le variabili dei moltiplicandi sono state dichiarate come `short int`, ma nel primo `scanf` viene utilizzata la notazione "%f", che va utilizzata per una variabile di tipo `float`, quindi sbagliata per un numero intero.

Errore n°3B^[4]: L'utilizzo della variabile `short int` permette l'input di un numero compreso tra -32768 a +32767, nonostante non sia necessariamente sbagliato, c'è il rischio che il prodotto sia un numero superiore o inferiore agli estremi permessi, incorrendo nella possibilità di un buffer overflow.

```
void dividi ()
{
    int a,b = 0;
    printf ("Inserisci il numeratore:");
    scanf ("%d", &a);
    printf ("Inserisci il denominatore:");
    scanf ("%d", &b);

    int divisione = a % b;

    printf ("La divisione tra %d e %d è: %d", a,b,divisione);
}
```

Errore n°4A^[5]: Nella funzione "dividi" viene utilizzato l'operatore modulo, che non è quello corretto per una divisione di numeri.

Errore n°4B^[6]: Anche se per "dividi" si volesse intendere una divisione con l'operatore modulo restituendo quindi il resto dell'operazione, il programma comunque non tiene conto della possibilità che l'utente potrebbe inserire come denominatore "0", correndo il rischio di incorrere in un errore.

```
void ins_string ()
{
    char stringa[10];
    printf ("Inserisci la stringa:");
    scanf ("%s", &stringa);
}
```

Errore n°5A^[7]: Nella funzione "ins_string" il vettore dichiarato è di 10 caratteri, ma all'utente questo limite non viene comunicato, e soprattutto non c'è alcun tipo di controllo sulla dimensione che l'input può avere, incorrendo ancora una volta nel rischio di buffer overflow qualora la stringa inserita fosse maggiore di 9 caratteri.

Errore n°5B^[8]: La notazione usata per `scanf` ("%s", &stringa) non è corretta perché "stringa" è stata dichiarata come un array, quindi l'utilizzo di "&" fa sì che `scanf` riceva l'indirizzo di memoria di "stringa", ma essendo "stringa" un array è già un indirizzo di memoria, si va quindi effettivamente a comunicare l'indirizzo dell'indirizzo di memoria. In questo modo `scanf` rischia di scrivere nella memoria in una posizione sbagliata, già occupata da altre variabili o causare altri problemi.

Errore n°6: In tutti i casi in cui viene utilizzato `scanf` non viene inserito lo spazio tra " e % causando quindi un incorretto funzionamento del programma che non permette all'utente di inserire un input, ma continua con le istruzioni successive.

```
GNU nano 7.2                                s2l5.c
1 #include <stdio.h>
2
3 void menu();
4 void moltiplica();
5 void dividi();
6 void dividim();
7 void ins_string();
8
9 int main() {
10     char scelta = {'\0'};
11     menu();
12     scanf (" %c", &scelta);
13
14     switch (scelta){
15         case 'A':
16             moltiplica();
17             break;
18         case 'B':
19             dividi();
20             break;
21         case 'C':
22             ins_string();
23             break;
24         case 'D':
25             dividim();
26             break;
27         default:
28             printf("Mi dispiace, la scelta non è valida\n");
29             break;
30     }
31
32     return 0;
33 }
34
```

```
GNU nano 7.2                                s2l5.c
35 void menu () {
36     printf("Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti\n");
37     printf("Come posso aiutarti?\n");
38     printf("A >> Moltiplicare due numeri \nB >> Dividere due numeri \nC >> Inserire una stringa \nD >> Resto tra due numeri: ");
39 }
40
41 void moltiplica () {
42     int a,b = 0;
43     printf("Inserisci i due numeri da moltiplicare: \n");
44     scanf(" %d", &a);
45     scanf(" %d", &b);
46
47     int prodotto = a * b;
48     printf ("Il prodotto tra %d e %d è: %d ", a,b,prodotto);
49 }
50
51 void dividi () {
52     int a,b = 0;
53     printf ("Inserisci il numeratore: ");
54     scanf (" %d", &a);
55     printf ("Inserisci il denominatore: ");
56     scanf (" %d", &b);
57
58     if (b == 0)
59     {
60         printf("Impossibile dividere per zero.\n");
61         return;
62     }
63
64     int divisione = a / b;
65     printf ("La divisione tra %d e %d è: %d", a,b,divisione);
66 }
67
68
```

```
63     int divisione = a / b;
64     printf ("La divisione tra %d e %d è: %d", a,b,divisione);
65 }
66
67
68
69 void dividim () {
70     int a,b = 0;
71     printf ("Inserisci il numeratore: ");
72     scanf ("%d", &a);
73     printf ("Inserisci il denominatore: ");
74     scanf ("%d", &b);
75
76     if (b == 0)
77     {
78         printf("Impossibile dividere per zero.\n");
79         return;
80     }
81
82     int divisionem = a % b;
83     printf ("Il resto della divisione tra %d e %d è: %d", a,b,divisionem);
84 }
85
86 void ins_string () {
87     char stringa[10];
88     printf ("Inserisci la stringa: ");
89     scanf ("%9s", stringa);
90 }
91
```

Line 12^[9]: Utilizzando "%c" che è la notazione riservata per i caratteri invece di "%d" che va utilizzata per i numeri interi, la scansione funziona e l'input dell'utente viene compreso e salvato.

Lines 27-29^[10]: Attraverso l'inserimento di un caso di default che va a coprire tutti gli input diversi da quelli elencati abbiamo ora gestito anche questa possibilità.

Line 44^[11]: Sostituendo la notazione sbagliata "%f" con quella giusta "%d" riservata per i numeri interi il programma interpreta correttamente l'input dell'utente.

Lines 42-47^[12]: Le variabili dei moltiplicandi e del prodotto sono state sostituite con `int` in modo da non rischiare più un buffer overflow in caso di numeri o prodotti maggiori a 32,767.

Line 64^[13]: Al posto dell'operatore modulo c'è ora "/" che è il corretto operatore per le divisioni tra numeri. Non essendo chiaro dal programma se l'obiettivo fosse la divisione tra due numeri o l'utilizzo dell'operatore modulo ho creato una quarta scelta (D) identica alla divisione tra due numeri ma che appunto fa uso dell'operatore modulo e fornisce il resto tra i due numeri inseriti.

Lines 58-62 & 76-80^[14]: Sia per la funzione "dividi" che per quella creata "dividim" è stata inserito un piccolo condizionale `if` che si occupa di controllare se il numeratore è 0, nel cui caso informa l'utente che ciò non è possibile.

Line 89^[15]: Attraverso l'inserimento del numero 9 tra % e "s" ora possiamo essere sicuri di non correre il rischio di buffer overflow qualora l'utente inserisse una stringa maggiore o uguale a 10 caratteri, e rimuovendo "&" risolviamo anche i potenziali conflitti nell'allocazione della memoria.

Lines ~: In tutto il programma sono stati inseriti spazi durante l'utilizzo di `scanf` per permettere un corretto funzionamento del codice, sono stati inoltre aggiunti vari `\n` per far sì che una volta compilato il programma fosse effettivamente leggibile e funzionale.

Di seguito è allegata la dimostrazione che il codice ora funziona e anche nei casi particolari.

```
kali@kali: ~/Desktop/Esercizi C/S2-L5
File Actions Edit View Help

(kali@kali)-[~/Desktop/Esercizi C/S2-L5]
$ ./s2l5
Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti
Come posso aiutarti?
A >> Moltiplicare due numeri
B >> Dividere due numeri
C >> Inserire una stringa
D >> Resto tra due numeri: A
Inserisci i due numeri da moltiplicare:
141245
62452
Il prodotto tra 141245 e 62452 è: 231098148

(kali@kali)-[~/Desktop/Esercizi C/S2-L5]
$ ./s2l5
Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti
Come posso aiutarti?
A >> Moltiplicare due numeri
B >> Dividere due numeri
C >> Inserire una stringa
D >> Resto tra due numeri: B
Inserisci il numeratore: 130
Inserisci il denominatore: 10
La divisione tra 130 e 10 è: 13

(kali@kali)-[~/Desktop/Esercizi C/S2-L5]
$ ./s2l5
Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti
Come posso aiutarti?
A >> Moltiplicare due numeri
B >> Dividere due numeri
C >> Inserire una stringa
D >> Resto tra due numeri: B
Inserisci il numeratore: 560
Inserisci il denominatore: 0
Impossibile dividere per zero.
```

```
(kali@kali)-[~/Desktop/Esercizi C/S2-L5]
$ ./s2l5
Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti
Come posso aiutarti?
A >> Moltiplicare due numeri
B >> Dividere due numeri
C >> Inserire una stringa
D >> Resto tra due numeri: C
Inserisci la stringa: esempio

(kali@kali)-[~/Desktop/Esercizi C/S2-L5]
$ ./s2l5
Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti
Come posso aiutarti?
A >> Moltiplicare due numeri
B >> Dividere due numeri
C >> Inserire una stringa
D >> Resto tra due numeri: D
Inserisci il numeratore: 35
Inserisci il denominatore: 6
Il resto della divisione tra 35 e 6 è: 5
```

1. E1 ↩
2. E2 ↩
3. E3A ↩
4. E3B ↩
5. E4A ↩
6. E4B ↩
7. E5 ↩
8. E5 ↩
9. E1 ↩
10. E2 ↩
11. E3A ↩
12. E3B ↩
13. E4A ↩
14. E4B ↩
15. E5 ↩