Consegna 2 febbraio S2L5

Dato il codice in allegato
- Capire cosa fa il programma senza eseguirlo.
- Individuare dal codice sorgente le casistiche non standard che il programma non gestisce (esempio, comportamenti potenziali che non sono stati contemplati).
- Individuare eventuali errori di sintassi / logici.
- Proporre una soluzione per ognuno di essi.

Il codice simula un'assistente digitale che svolge alcune funzionalità molto semplici come moltiplicare due numeri interi, dividere due numeri interi e da all'utente la possibilità di inserire una stringa di caratteri. Il codice contiene degli errori di sintassi , di logica e casistiche non standard che il programma non gestisce. Prima andrò a spiegare in linea generale come funziona il codice, poi andremo ad individuare gli errori.
Il codice:
Dopo la direttiva di compilazione, vengono dichiarate le funzioni che verranno definite successivamente. La funzione svolge un ruolo specifico, come nella caso di #void dividi ( ) che ci permette di dividere due numeri interi inseriti dall'utente.
La funzione Main è il punto di ingresso del programma. Inizializza una variabile di tipo char che potrà contenere la scelta dell'utente.Successivamente verrà chiamata la funzione menu ( ) che stampa un messaggio di benvenuto e attraverso lo switch presenta all'utente 3 opzioni disponibili (moltiplicare, dividere, inserire un stringa). All'interno dei casi di switch troviamo il richiamo alla funzione che viene sviluppata fuori dalla funzione main.
Fuori dalla funzione principale, troviamo la funzione menù, che ho già spiegato sopra. La funzione moltiplica () chieda all'utente di inserire due numeri interi (a,b), li moltiplica e ne stampa il risultato. La funziona dividi () chiede di inserire due numeri interi (a,b), li divide e li moltiplica. La funzione ins_string () chiede all'utente di inserire una stringa di massimo 10 caratteri.

Riporto di sotto il codice, ho commentato in rosso gli errori trovati.

```
1.  #include <stdio.h>
2.
3.  void menu ();
4.  void moltiplica ();
5.  void dividi ();
6.  void ins_string();
7.
8.  int main () {
9.
10.     char scelta = {'\0'}; // Rimuovere le graffe, non sono necessarie con le variabili di tipo char. Errore di logica
11.
12.     menu ();
13.     scanf ("%d", &scelta);//errore nel tipo di argomento, che è uno char, quindi %c. Andrebbe anche messo uno spazio prima di %c. Errore di logica
14.
15.     switch (scelta) {
16.
17.             case 'A':
18.             moltiplica();
19.             break;
20.             case 'B':
21.                 dividi();
22.                 break;
23.             case 'C':
24.                 ins_string();
```

```
25.                     break;
26.   //In questo casa manca il default, che non è obbligatorio ma  l'utente potrebbe inserire un valore diverso dalle variabilii disponibili. Questo per esempio è un caso che il programma non gestisce. Per esempio potremmo scrivere default printf("Non posso elaborare la richiesta, ritentano le scelte disponibili.\n); break; Errore logico
27.       }
28.
29. return 0;
30. }
31.
32. void menu ()
33. {
34.       printf ("Benvenuto, sono un assitente digitale, posso aiutarti a sbrigare alcuni compiti\n");// Errore grammaticale dell'programmatore
35.       printf ("Come posso aiutarti?\n");
36.       printf ("A >> Moltiplicare due numeri\nB >> Dividere due numeri\nC >> Inserire una stringa\n");
37.
38. }
39. void moltiplica ()
40. {
41.       short int  a,b = 0; // Con short int possiamo usare solo 2 di 16 byte, quindi potremmo avere problemi se l'utente inserisce numeri grandi
42.       printf ("Inserisci i due numeri da moltiplicare:");
43.       scanf ("%f", &a); //Errore logico , variabile di tipo shirt int quindi andrà inserito  %hd
44.       scanf ("%d", &b);// %hd Errore logico
45.
46.       short int prodotto = a * b;
47.
48.       printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto); // Errore logico , inserire %hd %hd %hd in quanto si riferiscono a  variabili di tipo short int
49. }

50.   void dividi ()
51. {
52.         int  a,b = 0; // Sarebbe corretto aggiungere int b!=o , il denominatore di una divisione non può essere 0.
53.         printf ("Inserisci il numeratore:");
54.         scanf ("%d", &a);
55.       printf ("Inserisci il denumeratore:");
56.         scanf ("%d", &b);
57.
58.         int divisione = a % b;// Operatore aritmetico sbagliato, va usato il simbolo di divisione ( / ).Errore di logico
59.
60.         printf ("La divisione tra %d e %d e': %d", a,b,divisione);
61. }
62.
63. void ins_string ()
64. {
65.       char stringa[10];//char stringa andrebbe inizializzato a zero ={'0'}. Avendo solo 10 caratteri disponibili  il numeri dei caratteri della  stringa inserita dall'utente potrebbero non rientrare nello spazio di memoria allocato, potremmo quindi andare incontro ad un buffer overflow. Siamo di fronte ad un caso di codice vulnerabile, un eventuale attaccante potrebbe inserire un codice dannoso e compromettere la sicurezza del sistema.
66.         printf ("Inserisci la stringa:");
67.         scanf ("%s", &stringa);// Non è necessario inserire l'operatore &  per leggere una stringa con 'scanf'. L'operatore & si  usa con tipi di dati che richiedono l'indirizzo di memoria, come interi o float
68. }
```