Venerdì 03/02/2023

Risposte alla traccia:

PUNTO1. Il codice in allegato rappresenta un semplice programma in linguaccio C che offre 3 opzioni all'utente tramite un menù: moltiplicare due numeri, dividere due numeri ed inserire una stringa.

La funzione "menu ()" stampa il menù di scelta dell'utente, la funzione "moltiplica ()" permette all'utente di inserire due numeri e stampa il prodotto di questi due, la funzione "dividi ()" permette all'utente di dividere un numeratore per un denominatore, infine la funzione "ins_string ()" permette all'utente di inserire una stringa di caratteri.

Il codice utilizza la funzione "switch" per eseguire l'azione scelta dall'utente.

PUNTO2. Casistiche non standard che il programma non gestisce:

- Il programma non controlla se l'input dell'utente sia valido. Ad esempio se l'utente inserisce una lettera diversa da A B o C il programma genera un errore e non permette di uscire dal programma. **Soluzione**: aggiungo "default: printf() break;" per indicare all'utente una scelta non valida, "while(); return 0;" dove all'interno di while andrò ad inserire tutte le scelte diverse da quelle proposte nel menu

- Sia nella moltiplicazione che nella divisione, il programma non gestisce l'inserimento di lettere o caratteri speciali. **Soluzione**: utilizzo funzione fgets per leggere una riga di testo completa come input, quindi utilizzare la funzione sscanf per analizzare il testo in base ai formati previsti.

- Il programma non pone un limite al massimo dei caratteri inseribili. **Soluzione**: inserire una costante globale #define MAX_LEN

- La funzione "dividi ()" non gestisce il caso in cui l'utente inserisca 0 come denominatore, il che causerà un errore di divisione per zero. **Soluzione**: inserisco un "if (denominatore == 0) {printf("Error: division by zero.\n");}

- La funzione "moltiplica()" utilizza il tipo di dati "short int" per le variabili a e b, questo significa che i valori possono essere tra -32768 e 32767, se si inserisce un numero fuori da questo intervallo l'operazione di moltiplicazione potrebbe causare un overflow. **Soluzione**: invece di "short int" utilizzo "int".

- La funzione "ins_string ()" non è sicura in quanto non limita la lunghezza della stringa inserita dall'utente. Questo ptorebbe causare un buffer overflow se l'utente inserisce una stringa troppo lunga. **Soluzione**: utilizzo la funzione fgets per leggere l'input dallo standard input (stdin).

- Dopo aver concluso un'operazione il programma si chiude, senza ripresentare all'utente la possibilità di scegliere nuovamente un'operazione dal menu. **Soluzione**: si può utilizzare un ciclo while che continui a ripetere il menu finché l'utente non decide di uscire.

-Nelle funzioni "moltiplica" e "dividi" non sono consentini numeri decimali. **Soluzione**: uso la funzione

"float" %f al posto di %d.

PUNTO 3. errori di sintassi e logici con relativa correzione:

- Nella funzione "main", lo switch statement sta confrontando una variabile di tipo char con un valore intero invece dovrebbe confrontarlo con una lettera.

- Nella funzione "dividi", la variabile "divisione" è calcolata come modulo tra due numeri (a % b), ma dovrebbe essere calcolata come divisione (a / b).

- Nella funzione "ins_string", la scanf viene utilizzata con l'indirizzo della stringa invece che con il nome della stringa. Dovrebbe essere cambiato in "%s".

PUNTO 4. Le correzioni sono state inserite nei vari punti dove si segnala l'errore o la situazione non gestita.