BUG HUNTING

Gentile Professore,

Ho effettuato un'analisi approfondita del programma fornito, con l'obiettivo di individuare eventuali problematiche e fornire delle proposte di soluzione. Di seguito, riporto i risultati della mia analisi:

1) **Capire cosa fa il programma senza eseguirlo:**

- Il programma è un assistente digitale che offre tre opzioni all'utente: moltiplicare due numeri, dividere due numeri o inserire una stringa.

- Il programma richiede all'utente di inserire una scelta **(carattere 'A', 'B' o 'C')** e poi esegue la funzione corrispondente in base alla scelta effettuata.

- La funzione **moltiplica()** chiede all'utente di inserire due numeri interi, li moltiplica e restituisce il risultato.

- La funzione **dividi()** chiede all'utente di inserire due numeri interi, effettua la divisione e restituisce il risultato.

- La funzione **ins_string()** chiede all'utente di inserire una stringa di massimo 9 caratteri e la stampa.

- Dopo l'esecuzione dell'operazione scelta, il programma termina.

2) **Individuare dal codice sorgente le casistiche non standard che il programma non gestisce:**

- Il programma non gestisce il caso in cui l'utente inserisce una scelta diversa da 'A', 'B' o 'C' nel menù. Non è prevista alcuna azione specifica per gestire questa situazione.

3) **Individuare eventuali errori di sintassi/logici:**

Errori di sintassi:

- La dichiarazione del carattere **scelta** nella funzione **main()** è inizializzata come un array di caratteri **{}** invece di un singolo carattere **"**.

- Nella funzione **ins_string()**, l'argomento per la funzione **scanf()** per leggere una stringa utilizza l'operatore di indirizzo **&**, che non è necessario.

Errori logici:

- Nella funzione **moltiplica()**, la variabile **a** viene letta come un **float** anziché come un intero.

- Nella funzione **moltiplica()**, la variabile **b** viene letta come un **intero**, ma viene dichiarata come **short int**.

- Nella funzione **dividi()**, l'operazione di divisione utilizza l'operatore modulo **%** anziché l'operatore di divisione **/**.

4) **Proporre una soluzione per ognuno di essi:**

- Nella funzione **main()**, correggere la dichiarazione del carattere **scelta** come **char scelta = '\0';**.

- Nella funzione **moltiplica()**, correggere la lettura della variabile **a** come un intero anziché un float e correggere la dichiarazione della variabile **b** come int invece di **short int**.

- Nella funzione **dividi()**, correggere l'operazione di divisione utilizzando l'operatore **/** anziché **%**.

- Nella funzione **ins_string()**, rimuovere l'operatore di indirizzo **&** dall'argomento della funzione **scanf()** per leggere la stringa.

Report di analisi del programma:

Il programma che ho analizzato è un'applicazione che presenta un menu interattivo all'utente con tre opzioni: moltiplicare due numeri, dividere due numeri o inserire una stringa. L'utente può selezionare una delle opzioni inserendo una lettera corrispondente al menù.

Durante l'analisi del codice sorgente, sono state individuate alcune problematiche che possono influire sul corretto funzionamento del programma:

Gestione delle scelte non valide: Il programma non gestisce il caso in cui l'utente inserisce una scelta diversa da 'A', 'B' o 'C' nel menu. Non è prevista alcuna azione specifica per gestire questa situazione, il che potrebbe portare a risultati imprevisti o a un'interruzione del programma. Si consiglia di implementare una gestione degli input non validi, ad esempio con un messaggio di errore e un invito a reinserire la scelta corretta.

Durante l'analisi del codice sorgente, sono stati individuati anche alcuni errori di sintassi e logici, che richiedono correzioni:

Dichiarazione errata del carattere scelta: Nella funzione **main()**, la dichiarazione del carattere scelta è inizializzata come un **array** di caratteri **{}** invece di un singolo carattere **''**. Per correggere questo errore, la dichiarazione corretta del carattere scelta dovrebbe essere **char scelta = '\0'**;.

Lettura errata della variabile a nella funzione moltiplica(): Nella funzione **moltiplica()**, la variabile a viene letta come un **float** anziché come un **intero**. Per correggere questo errore, la lettura della variabile **a** dovrebbe essere corretta per leggere un **intero**.

Dichiarazione inconsistente della variabile b nella funzione moltiplica(): Nella funzione **moltiplica()**, la variabile **b** viene letta come un **intero**, ma viene dichiarata come **short int**. Questo potrebbe causare errori di **overflow o underflow**. Si consiglia di correggere la dichiarazione della variabile **b** come int invece di **short int**.

Operatore errato nella funzione dividi(): Nella funzione **dividi()**, l'operazione di divisione utilizza l'operatore modulo **%** anziché l'operatore di **divisione /**. Per ottenere il risultato corretto della divisione, si consiglia di correggere l'operazione di divisione utilizzando **l'operatore /**.

Uso errato dell'operatore di indirizzo nella funzione ins_string(): Nella funzione **ins_string()**, l'argomento per la funzione **scanf()** per leggere una stringa utilizza l'operatore di indirizzo **&**, che non è necessario. Per correggere questo errore, l'operatore di indirizzo **&** deve essere rimosso dall'argomento della funzione **scanf()** per leggere correttamente la stringa.

CONCLUSIONI:

In conclusione, il programma che ho analizzato presenta un menu interattivo che permette all'utente di selezionare diverse operazioni. Tuttavia, sono state identificate alcune problematiche che richiedono correzioni per garantire il corretto funzionamento del programma. Queste problematiche includono la gestione delle scelte non valide, errori di sintassi come la dichiarazione errata del carattere scelta e la lettura errata di variabili, e errori logici come l'utilizzo sbagliato degli operatori di divisione e indirizzo.

Per migliorare il programma, si raccomanda di implementare una gestione degli input non validi nel menu, fornendo un messaggio di errore e richiedendo all'utente di reinserire una scelta corretta. È anche importante correggere gli errori di sintassi, come la dichiarazione corretta del carattere scelta e la lettura corretta delle variabili. Inoltre, è necessario utilizzare l'operatore corretto per l'operazione di divisione e rimuovere l'operatore di indirizzo non necessario per la lettura della stringa.

Queste correzioni aiuteranno a garantire un funzionamento corretto del programma, migliorando l'esperienza utente e riducendo la possibilità di errori. È sempre importante prestare attenzione alla gestione degli input non validi e assicurarsi che le operazioni siano eseguite correttamente per ottenere i risultati desiderati.