Giorno10-Progetto: Bug Hunting EPICODE

Giovanni Pisapia

Capire come funziona il programma senza eseguirlo:

1. Viene importata la libreria tramite #include <stdio.h>.
2. Vengono dichiarate le funzioni che aiuteranno a mantenere il codice pulito e di facile lettura.
3. In main() viene dichiarata una variabile di tipo char chiamata scelta, con valore inizializzato a zero, e viene richiamata la funzione menu(). Successivamente, tramite lo scanf(), viene acquisito l'input dell'utente e memorizzato nella variabile scelta.
4. Viene utilizzato uno switch per eseguire la funzione corrispondente alla scelta dell'utente, con un break alla fine di ogni caso per interrompere lo switch.
5. La funzione void menu() mostra una breve descrizione del programma e la scelta disponibile, quindi la funzione viene richiamata nuovamente per consentire all'utente di fare un'altra scelta.
6. Una volta che l'utente ha fatto la sua scelta, verrà avviato il processo switch.
7. La funzione void moltiplica() richiede all'utente di inserire due numeri tramite la printf(), che verranno memorizzati nelle variabili a e b tramite lo scanf(). Viene creata una nuova variabile di tipo short int chiamata prodotto, il cui valore sarà uguale alla moltiplicazione di a*b inseriti dall'utente. Il risultato verrà stampato tramite la printf(), che prenderà i valori dallo scanf() e dalla variabile prodotto.
8. La funzione void divisione() funzionerà in modo simile alla precedente, con la differenza che verrà utilizzata la variabile di tipo short int chiamata divisione, che utilizzerà l'operatore % per trovare il resto della divisione. Il risultato verrà stampato tramite la printf(), che stamperà il valore inserito dall'utente tramite lo scanf() e il valore della variabile divisione.
9. Infine, nella funzione void ins_string(), verrà richiesto all'utente di inserire una stringa di massimo 10 caratteri, senza però restituire alcun valore.

Errori Trovati nel codice e come correggerli

1. Nella dichiarazione della variabile "scelta" nel main, le parentesi graffe non sono necessarie.
2. Nel secondo scanf del main, è usato il simbolo %d per la variabile "scelta" che è stata dichiarata come carattere (char), pertanto il formato corretto dovrebbe essere %c.
3. Nella funzione "moltiplica", la variabile "a" viene scansionata con il formato %f invece di %d, che è quello corretto per gli interi. Inoltre, non ci sono controlli sul tipo di carattere inserito dall'utente.
4. Nella funzione dividi c'è un errore grammaticale denumeratore invece denominatore
5. Nella funzione "dividi", il simbolo utilizzato per la divisione è il resto % invece di /, che è il simbolo corretto. Inoltre, non ci sono controlli sul tipo di carattere inserito dall'utente.
6. Non ci sono controlli sulla lunghezza della stringa inserita dall'utente nella funzione "ins_string", il che può portare a un buffer overflow se viene inserita una stringa più lunga di 10 caratteri.
7. La variabile "stringa" è già un puntatore all'inizio dell'array, quindi non è necessario usare l'operatore & nell'argomento di scanf().
8. Non viene stampato a video il risultato della stringa inserita dall'utente nella funzione "ins_string".

Implentazione codice

- Per migliorare il codice, potremmo inserire all'interno della funzione main() un ciclo while, in modo che l'utente debba reinserire la scelta nel caso in cui risponda con una lettera diversa dalle opzioni "a", "b" o "c". Inoltre, potremmo rimuovere la dichiarazione di short int per le variabili a e b.
- Per quanto riguarda la funzione dividi(), potremmo inserire un altro ciclo while in modo che l'utente debba inserire un numero diverso da zero. Inoltre, potremmo aggiungere un ulteriore controllo sui caratteri inseriti dall'utente, utilizzando la seguente porzione di codice: if (scanf("%d", &a) != 1), in modo che l'utente possa inserire solo numeri interi e non lettere o altri simboli.
- Potremmo apportare le stesse modifiche anche alla funzione moltiplica(), rimuovendo la dichiarazione di int short e utilizzando if (scanf("%d%d", &a, &b) != 2) per garantire che l'utente inserisca solo numeri interi e non lettere o altri simboli.
- Inoltre, potremmo modificare la dimensione dell'array stringa in modo da poter contenere una stringa di 10 caratteri più il terminatore \0. Infine, potremmo inserire un ciclo while all'interno della funzione leggiStringa() per richiedere all'utente di inserire una stringa valida, con un massimo di 10 caratteri.
-