

Lettura e comprensione del codice sbagliato

Il codice si presenta con la dichiarazione di 4 funzioni:

- void menu (): stampa un breve messaggio di benvenuto e dà le opzioni disponibili su quello che si vuole fare
- void moltiplica (): dichiara due variabili di tipo short int, poi legge l'input dall'utente dei numeri che si vogliono moltiplicare, ne esegue il prodotto e lo stampa a schermo
- void dividi (): dichiara due variabili di tipo int, poi legge l'input dall'utente dei numeri che si vogliono dividere
- void ins_string(): viene dichiarato un array di 10 caratteri (indice da 0 a 9), viene letta la stringa dall'utente e poi stampata.

Nel main() viene inizializzata una variabile scelta di tipo char a '\0', successivamente viene invocata la funzione menu() che stampa le opzioni disponibili per l'utente. Al ritorno della funzione menu() viene letto l'input dall'utente e in seguito viene fatto uno switch case delle opzioni disponibili in base alla scelta fatta che richiama ad altre funzioni, moltiplica(), dividi() o ins_string().

Osservazioni riguardo errori e comportamenti scorretti del codice

Il codice presenta diversi errori, elencati di seguito:

Nel main la variabile scelta è dichiarata scorrettamente utilizzando le parentesi graffe, nella prima scanf viene utilizzato %d ma la variabile "scelta" è di tipo char quindi bisognerebbe utilizzare %c. Nello switch-case manca un caso di default nel caso l'utente non inserisca nessuna delle scelte possibili, bisogna aggiungere un case "default".

Nella funzione moltiplica() la variabile "a" è stata dichiarata ma non inizializzata, inoltre nella stampa viene utilizzata una formattazione errata, se le variabili sono di tipo short bisogna usare %hd, se sono di tipo int %d. Nella scanf viene utilizzato %f ma la variabile a è di tipo short quindi va utilizzato %hd.

Nella funzione divisione() la variabile "a" è stata dichiarata ma non inizializzata e viene utilizzato l'operatore sbagliato "%" che viene utilizzato per calcolare il resto mentre l'operatore corretto da utilizzare è "/". Prima di effettuare la divisione bisognerebbe aggiungere un controllo che l'utente non inserisca un denominatore uguale a 0 e in questo caso stampare un errore altrimenti verrebbe generato un errore runtime, se invece viene inserito un denominatore corretto si esegue la divisione.

Nella funzione ins_stampa() non va utilizzato l'operatore & per le stringhe inoltre per evitare buffer overflow si potrebbe ampliare l'array della stringa ad esempio stringa[100] (indici da 0 a 99) e in fase di lettura input dall'utente aggiungere un controllo attraverso %99s che alla fine aggiunge un carattere newline in automatico raggiungendo quota 100 caratteri inseriti.