**BUG HUNTING**

Il seguente programma crea un assistente digitale in grado di svolgere alcuni compiti. Possiamo osservarli nelle 4 funzioni (“menu”, “moltiplica”, “dividi” e “ins\_string”) definite sotto l’inclusione della libreria standard “stdio.h” che ci permette di gestire stream di input e output come “scanf” o “printf”. La funzione “menu” stampa un messaggio di benvenuto e offre le opzioni di scelta disponibili. La funzione “moltiplica” ci consente di inserire due numeri e poi moltiplicarli, stampando il risultato. La funzione “dividi” ci permette di inserire due numeri, effettuare la divisione e stampare il quoziente. Infine la funzione “ins\_string” ci consente di inserire una stringa.

Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

Il programma presenta diversi errori:

* **SCANF NELLA RIGA 14**



Nella riga 14 che riguarda la scanf, “%d” non ci consente di leggere correttamente un tipo char poiché si riferisce a un tipo di dato intero (int). Andiamo a sostituirlo con “ %c” assicurandoci inoltre di includere uno spazio fra il simbolo delle virgolette e quello della percentuale, affinchè “scanf” ignori eventuali spazi bianchi in input prima di leggere il dato. In questo modo:

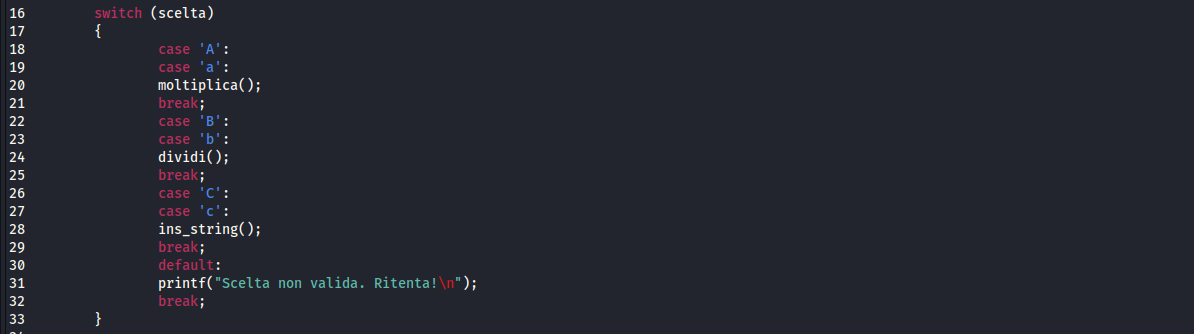


* **SWITCH SENZA OPZIONE DI DEFAULT**

Immagine che contiene testo, schermata

Descrizione generata automaticamente

Nello switch manca il caso di default, che ci permette di avere un’opzione da eseguire quando nessuna delle scelte presenti nei diversi casi (“case”) corrisponde alla scelta fatta. In questo caso infatti, senza l’opzione di default, lo switch terminerà senza eseguire alcun blocco di codice specifico. Scelgo nei vari casi di permettere anche l’inserimento delle lettere minuscole corrispondenti:

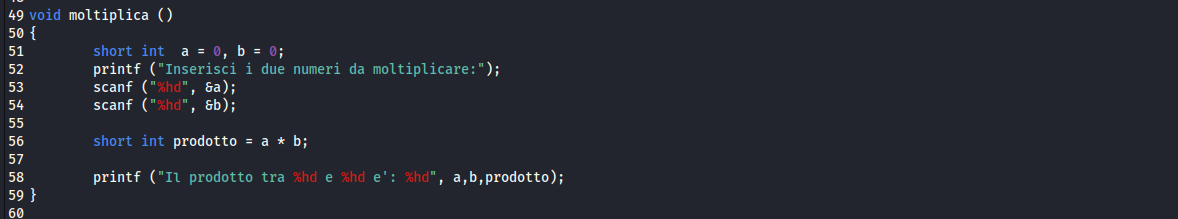


* **FUNZIONE “MOLTIPLICA”**

Immagine che contiene testo, schermata

Descrizione generata automaticamente

Qui troviamo un errore di gestione degli input. Innanzitutto, nella riga 45, solo la variabile b viene inizializzata. Occorre dunque inizializzare anche la variabile a separatamente. I due valori sono degli short int, ma nella scanf delle variabili invece troviamo: “%f” nella riga 47 che si riferisce a un valore float, e “%d” nella riga successiva che si riferisce a un valore int. Correggo anche quest’ultimo per evitare problemi di tipo o di dimensione delle variabili, sostituendoli con “%hd”. La funzione corretta si presenterà così:

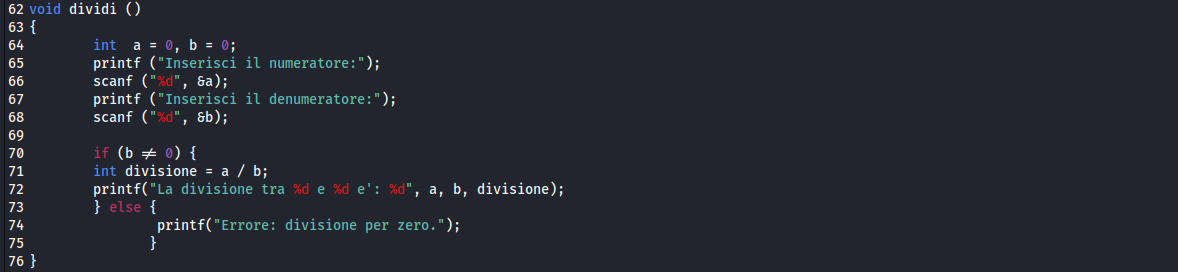


* **FUNZIONE “DIVIDI”**

Immagine che contiene testo, software, Software multimediale, schermata

Descrizione generata automaticamente

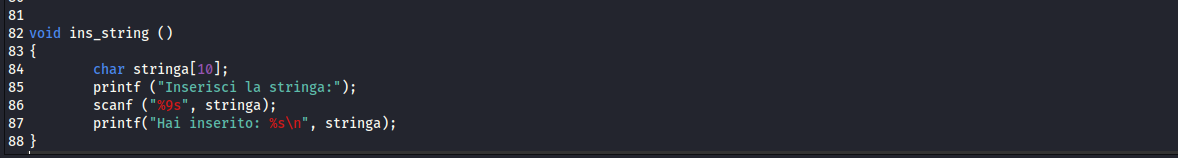
Anche nella funzione “dividi” ci assicuriamo nella riga 58 di inizializzare le due variabili separatamente. Qui possiamo notare l’uso sbagliato dell’operatore “%” che ci da come risultato il resto di una divisione, non la divisione stessa. L’operatore corretto in questo caso è “/”. Possiamo anche inserire un controllo per assicurarci che b non sia uguale a zero (la divisione non sarebbe possibile), inserendo l’operazione in un “if-else” e stampando un eventuale messaggio di errore. In questo modo:



* **FUNZIONE “INS\_STRING”**



Questa funzione acquisisce una stringa, ma la variabile è dichiarata con una dimensione fissa di 10. Se l’utente sceglie di inserire una stringa più lunga potrebbe verificarsi un buffer overflow. Pertanto vado a scrivere “%9s” nella scanf, controllando anche di aver rimosso il carattere “&” su “stringa”. Scelgo anche di stampare la stringa inserita con printf:



CODICE CORRETTO:Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamenteImmagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

Eseguiamo il programma:

