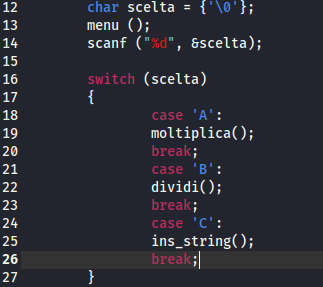
**Bug Hunting**

**Funzione del programma**

Come si evince dal codice, il programma fornito è un bot che ha il fine di aiutare l’utente in 3 compiti: moltiplicazioni, divisioni e inserimento di stringhe.

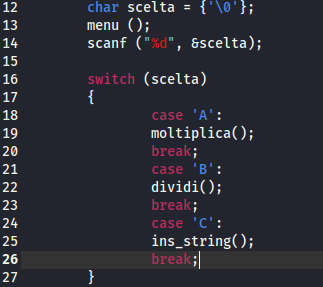
**Errori nel codice**

**Errore n°1**

****

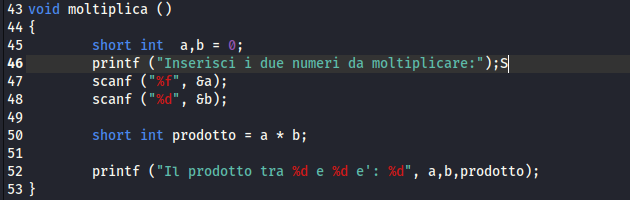
Alla riga 12 si è definita la variabile char scelta. ma alla riga 14 viene utilizzato l’operatore %d e non l'operatore %c che è quello corretto per la variabile char. Inoltre, non vi è il carattere di spazio tra apice e percentuale per far funzionare correttamente la scansione.

**Errore n°2**

****

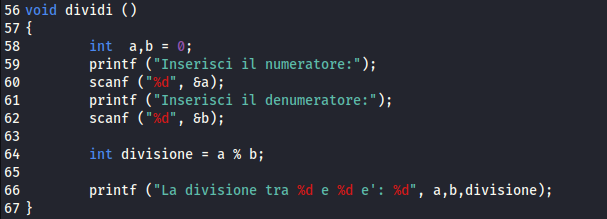
Alle righe 18,21 e 24, non si è tenuto conto delle opzioni in minuscolo come a,b,c.   
Poi alla riga 27 si è messa la parentesi graffa per la fine del ciclo switch. Questo però è un errore perché così il ciclo switch non ha il parametro di default e quindi non offre opzioni per quando l’utente immette un input non previsto dal ciclo. Non validare correttamente l’input dell’utente può portare ad effetti indesiderati o non sicuri nel programma come crash.

**Errore n°3**

****

Nella riga 45 si sono assegnate le variabili short int ad a e b ma nella riga 47, viene usato l’operatore %f della variabile float. La variabile giusta da mettere alle righe 45 e 50 è int per avere due numeri interi di 4 byte e non creare incompatibilità con il risultato del prodotto di due numeri short che potrebbe essere un numero intero più grande di due byte.

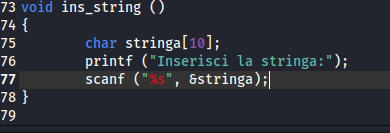
**Errore n°4**

****

Alla riga 64, si è utilizzato il simbolo % ma è sbagliato. Utilizzandolo, non otterremo il risultato della divisione ma il resto della stessa.

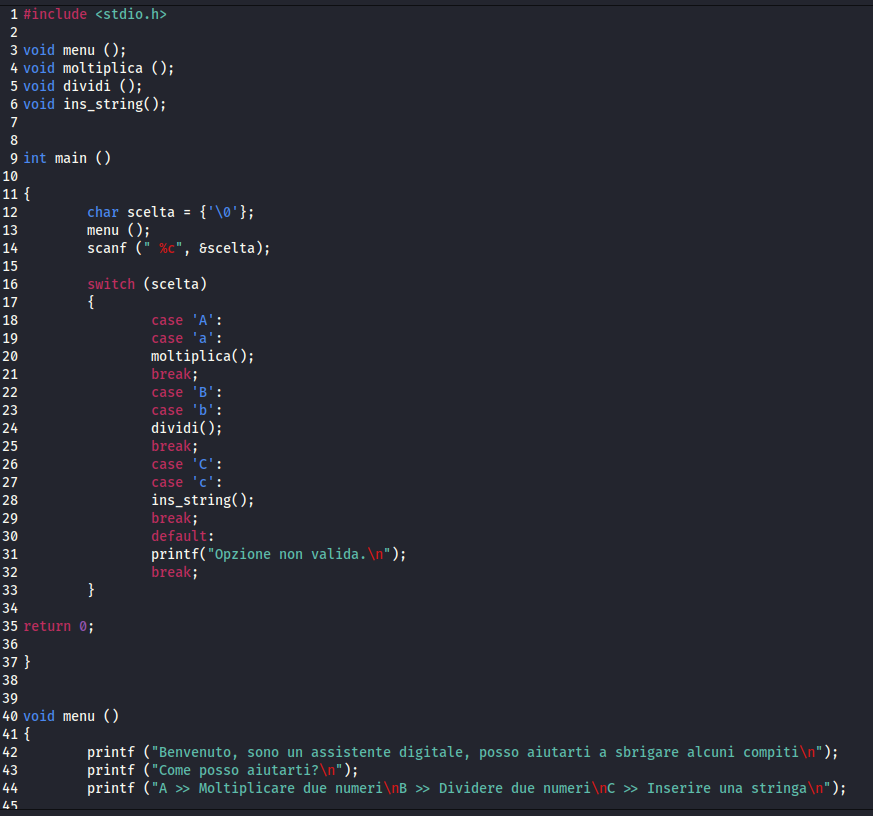
Inoltre non si è tenuto conto della divisione per 0, senza fornire alcuna limitazione a riguardo. Dividere per 0 è matematicamente impossibile quindi bisogna arginare il problema per evitare comportamenti anomali.

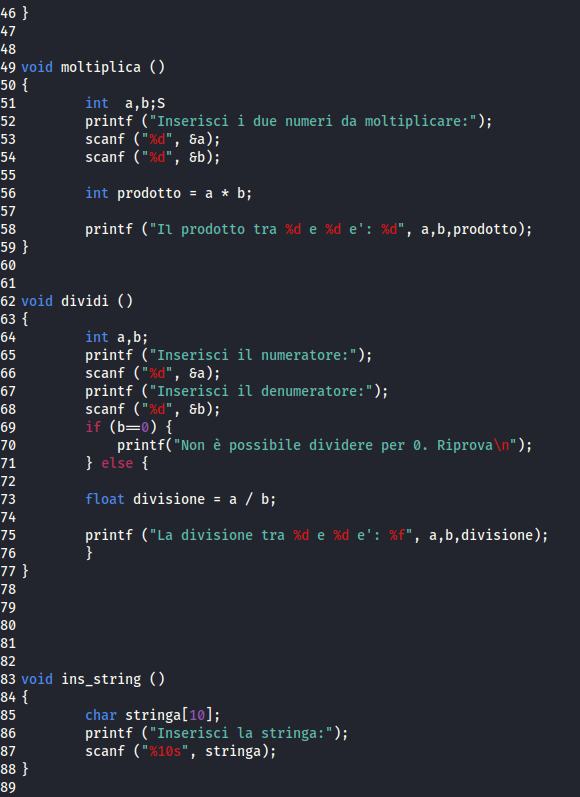
**Errore n°5**



Alla riga 77 , l’operatore %s è sbagliato.  
Questo perché non c’è controllo sull’input dell’utente nonostante si sia specificata una stringa di 10 caratteri nella variabile. Questa sarebbe un'occasione ghiotta per un malintenzionato perché potrebbe mandare in buffer overflow il sistema, scrivendo più caratteri di quelli previsti e occupando molta più memoria nel sistema fino a farla straripare.

**Correzione**

****



**Errore n°1**

Nella riga 14, ho messo %c per una corretta esecuzione, immettendo anche uno spazio tra apice e %c.

**Errore n°2**

Nelle righe 19,22,24 ho inserito la casistica delle lettere a, b , c in minuscolo.

Nelle righe 30 e 31 ho aggiunto l’opzione per il default per gli input non previsti dal programma.

**Errore n°3**

Nelle riga 45 ho assegnato la variabile int ad a e b. Nella riga 47, ho usato %d invece che %f. Nella riga 50 ho assegnato la variabile int al prodotto.

**Errore n°4**

Nelle righe 58 e 64, ho assegnato la variabile int sia ad a,b e la variabile float alla divisione, per riprodurre accuratamente i numeri reali. Sempre nella riga 64, ho modificato % con /, in modo tale da avere il risultato della divisione e non il suo resto. Infine nella riga 69 ho iniziato un ciclo if, nel caso b fosse uguale a 0, l’operazione verrebbe annullata e invalidata.

**Errore n°5**

Nella riga 87, ho modificato %s in %10s e ho tolto &.

Così facendo viene specificato il numero massimo di caratteri letti in input a 10, evitando possibili buffer overflow e rendendo più sicuro il programma.