**Progetto Bug Hunting**

**Funzionalità del programma**

Il programma è un assistente digitale molto basico sviluppato per fare due operazioni aritmetiche (Moltiplicazione, Divisione) e l’inserimento di una stringa.

Libreria usata <stdio.h> = "standard input-output header".

Sono stati creati le seguenti quattro funzioni di tipo **void e** dichiarate prima della funzione main**:**

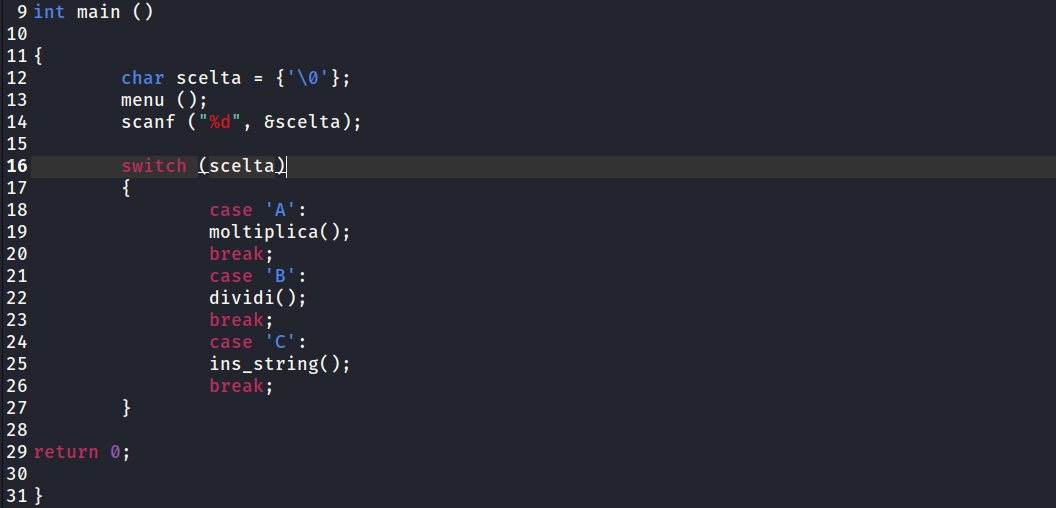
1. **menu**: Descrizione del programma.
2. **moltiplica**: Operazione aritmetica dei due valori inseriti dall’utente.
3. **dividi**: Operazione aritmetica dei due valori inseriti dall’utente.
4. **ins\_string**: Inserimento di una catena di caratteri.

Le funzioni vengono poi richiamate dentro la funzione principale **main()** per eseguire il programma.

**Errori dentro la funzione main**

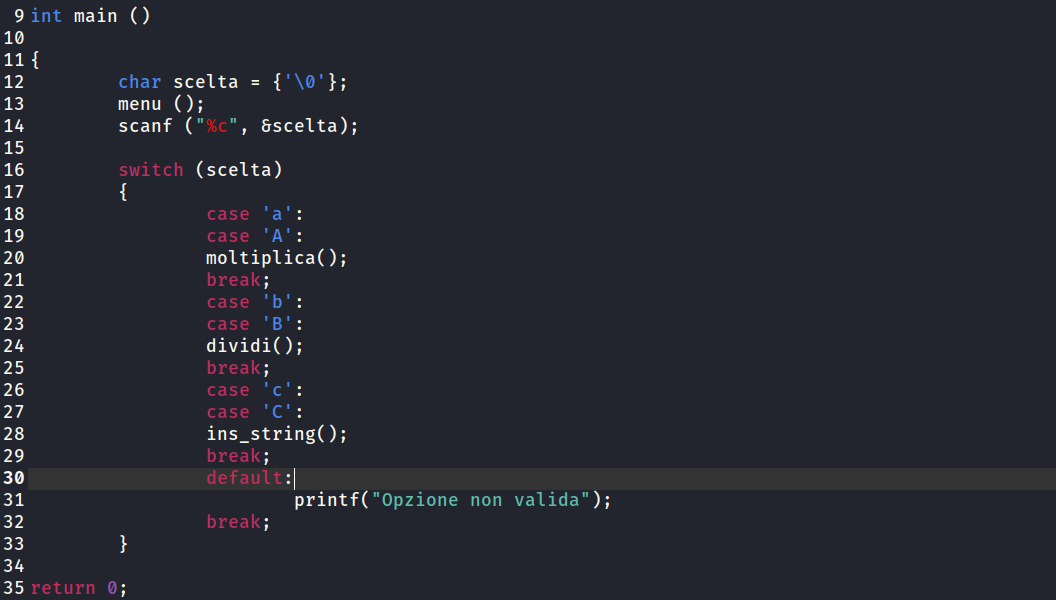
Dentro la funzione main possiamo trovare il primo errore sulla riga 14, è stato implementato un “format specifier” per un argomento di tipo int quando inserito dall’utente sarà di tipo char.(errore logico)

Per risolvere questo errore basta inserire il “format specifier” **%c** che viene usato per i char.



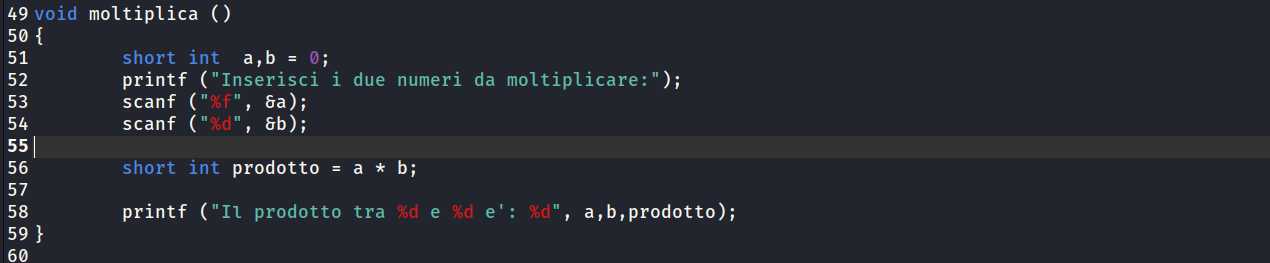
Per ottimizzare lo switch possiamo aggiungere altre 3 case per le lettere in minuscola o aggiungere un printf specificando all’utente di inserire la lettera in maiuscola e per correttezza sarebbe idoneo inserire un default dentro lo switch se l’opzione non è disponibile.

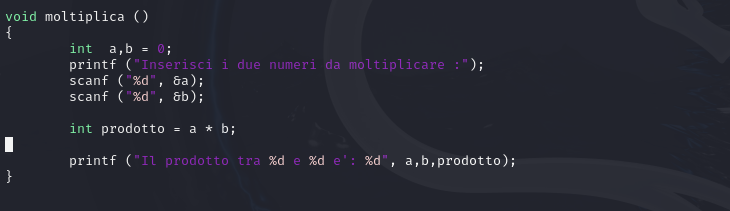
**Soluzione**



**Errori dentro la funzione moltiplica**

Abbiamo 2 variabili di tipo “short int” dichiarati e inizializzati uguale a 0, dato che l’operazione aritmetica delle due variabili possono superare il range di bytes del tipo dato “short int” inseriamo int per le variabili a , b.

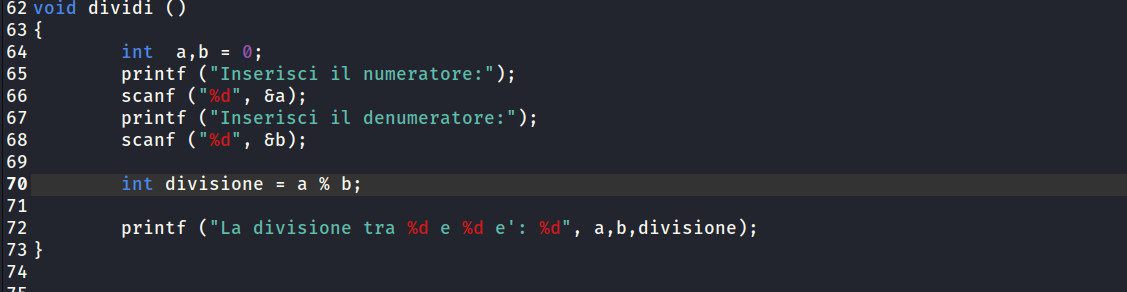


Per la scansione degli input dell’utente si deve cambiare il “format specifier” a %d per evitare un errore logico. 

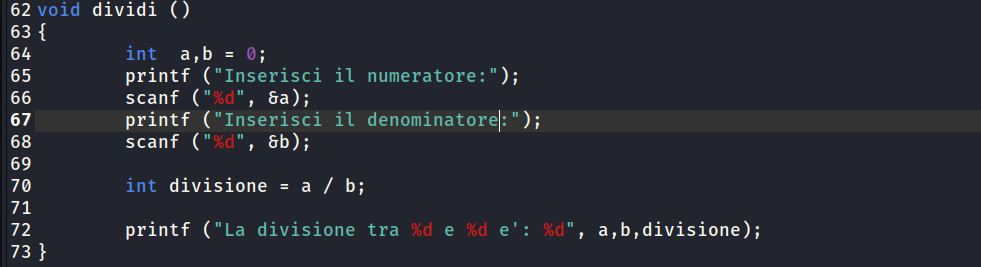
**Errori dentro la funzione dividi**

L’errore dentro questa funzione si trova sulla riga 70 al momento della inizializzazione della variable “divisione” l’operatore aritmetico da usare è ‘ / ‘ e non il modulus operator (%) perché ci ritorna il resto della divisione.

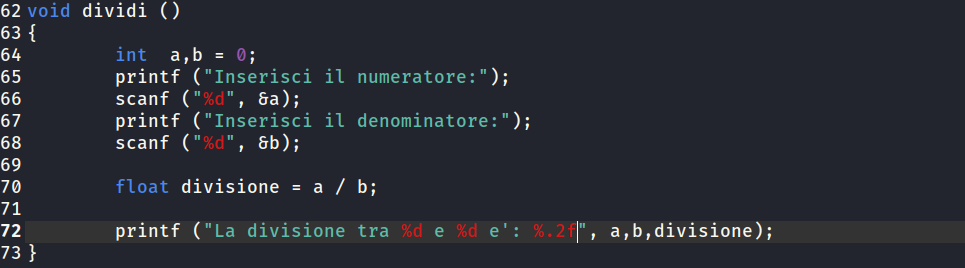
Modificare anche il printf() della riga 67 perché c’è un errore di grammatica “denumeratore”



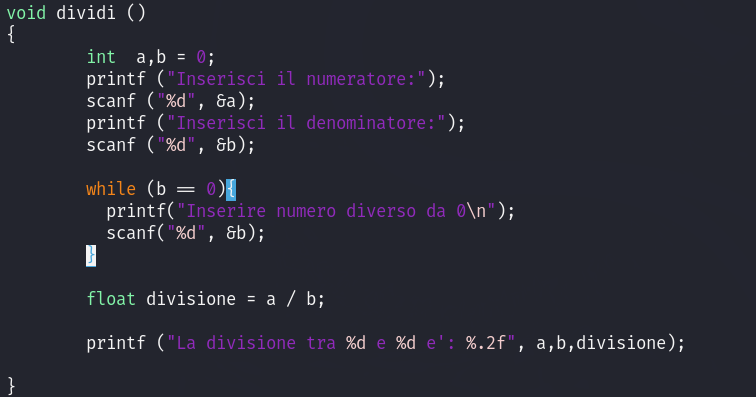
Per correggere questo errore basta solo cambiare il tipo di operatore e modificare la scritta della riga 67 “denominatore”.



Se vogliamo ottimizzare questa funziona possiamo cambiare il tipo di dato della variabile “divisione” a float e dentro la nostra funzione printf() modifichiamo l’ultimo format specifier a ‘ %.2f ’.



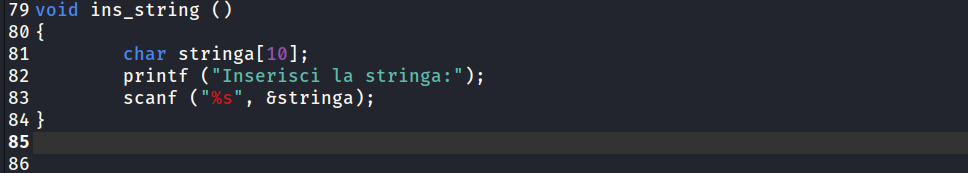
Ulteriore ottimizzazione sarebbe di aggiungere un controllo se il denominatore è uguale a 0, se è uguale a 0 il programma dovrà richiedere un nuovo denominatore.



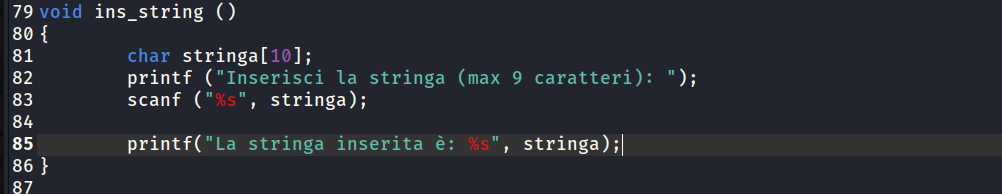
**Funzione ins\_string**

Questa funziona è un codice incompleto, l’utente può inserire una string (nome, cognome, parola, ecc) e salvarlo dentro la variabile char[] (character array) che conterrà ogni char inserito dall’utente creando una catena di caratteri chiamati anche string ma il nome inserito non verrà visualizzato.

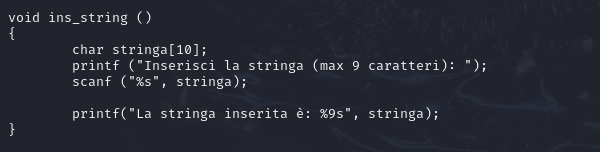
Sulla riga 83 dentro la funzione scanf() non è necessario usare il simbolo ‘&’ perché l’input inserito dall’utente andrà salvato dentro il character array che viene già trattato come un puntatore.



Per correggere questo basta solo aggiungere la funzione printf per visualizzare l’input e togliere il simbolo ‘&’ della riga 83.

Sulla riga 82 è necessario inserire il numero massimo di caratteri che accetta il nostro programma per evitare errori di runtime. 

Un’altra opzione sarebbe semplicemente leggere la lunghezza 9 dell'input inserito dall’utente dentro la seconda funzione printf().



**Codice ottimizzato**

#include <stdio.h>

void menu ();

void moltiplica ();

void dividi ();

void ins\_string();

int main ()

{

char scelta = {'\0'};

menu ();

scanf ("%c", &scelta);

switch (scelta)

{

case 'a':

case 'A':

moltiplica();

break;

case 'b':

case 'B':

dividi();

break;

case 'c':

case 'C':

ins\_string();

break;

default:

printf("Opzione non valida");

break;

}

return 0;

}

void menu ()

{

printf ("Benvenuto, sono un assitente digitale, posso aiutarti a sbrigare alcuni compiti\n");

printf ("Come posso aiutarti?\n");

printf ("A >> Moltiplicare due numeri\nB >> Dividere due numeri\nC >> Inserire una stringa\n");

}

void moltiplica ()

{

int a,b = 0;

printf ("Inserisci i due numeri da moltiplicare :");

scanf ("%d", &a);

scanf ("%d", &b);

int prodotto = a \* b;

printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto);

}

void dividi ()

{

int a,b = 0;

printf ("Inserisci il numeratore:");

scanf ("%d", &a);

printf ("Inserisci il denominatore:");

scanf ("%d", &b);

while( b == 0){

printf(“Inserire numero diverso da 0\n”);

scanf(“%d”, &b);

}

float divisione = a / b;

printf ("La divisione tra %d e %d e': %.2f", a,b,divisione);

}

void ins\_string ()

{

char stringa[10];

printf ("Inserisci la stringa (max 9 caratteri): ");

scanf ("%s", stringa);

printf("La stringa inserita è: %s", stringa);

}