Analisi del codice , ricerca e correzione degli errori dello stesso.

Ricevuto il codice, ho cercato di **analizzare** il codice sorgente per comprendere al meglio, la sua ***funzione***, gli ***errori presenti*** e come poterli ***risolvere***.
Qui di seguito andremo assieme ad analizzare ogni fase.

Il codice presentato ha come scopo o ***funzione*** quello di eseguire un applicativo che rende possibile la moltiplicazione, la divisione e l'inserimento di una stringa.
Questo dettaglio lo ricaviamo dalla struttura del codice direttamente nel ***`void menu`*** come riportato qua sotto in figura.

```
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");
```

Nel codice presentato con la sezione relativa allo switch l'unico caso che non contempla è il caso in cui l'utente che lo esegue, inserisce un valore diverso da A, B o C come è possibile vedere nella figura sotto.

```
int main ()

{
        char scelta = {'\0'};
        menu ();
        scanf ("%d", &scelta);

        switch (scelta)
        {
                case 'A':
                moltiplica();
                break;
                case 'B':
                dividi();
                break;
                case 'C':
                ins_string();
                break;
        }

return 0;

}
```

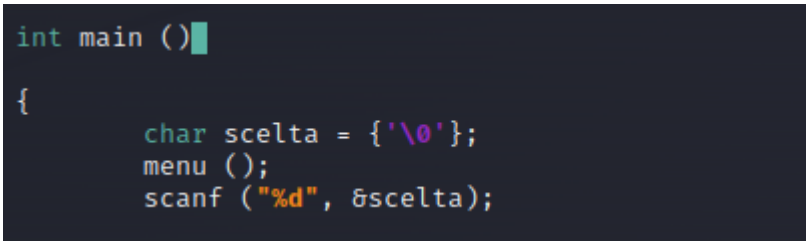
Per risolverlo potremmo dichiarare un altro caso chiamandolo subito dopo il ***`case 'C'`*** con le seguenti stringhe:

```
default:
            printf("Opzione non valida.\n");
            break;
```

Analisi del codice , ricerca e correzione degli errori dello stesso.

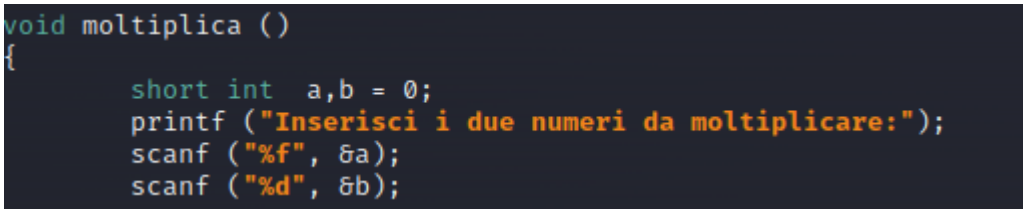
Il codice presentato presenta parecchi errori logici e di sintassi al suo interno ed adesso andremo a riconoscerli, individuarli e risolverli.

Partendo dall'alto notiamo subito con la figura sotto che nella dichiarazione della variabile scelta nella funzione ***`main()`***, sono state utilizzate le graffe invece delle virgolette singole per inizializzare il carattere vuoto.
La correzione che andremo a fare sarà per cui ***`char scelta = '\0'`*** invece di ***`char scelta = {'\0'}.`***



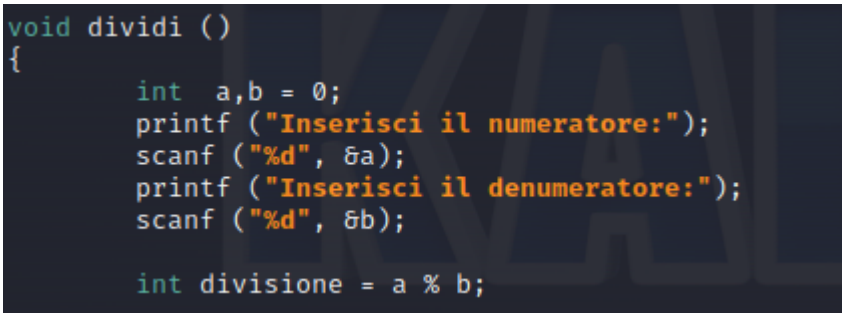
Scorrendo verso il basso possiamo notare con la figura che nella funzione ***`menu()`***, è stato utilizzato l'operatore di stampa ***`%d`*** per la variabile scelta nella chiamata a ***`scanf()`***.
Utilizziamo invece ***`%c`*** poiché scelta è un carattere.
La correzione sulla riga sarà per cui ***`scanf(" %c", &scelta),`*** questo lo facciamo per evitare problemi di buffer.

Ancora sotto nella funzione ***`moltiplica()`*** come possiamo vedere che vengono dichiarate le variabili a e b come short int, ma viene utilizzato lo specificatore di formato errato nella chiamata a ***`scanf()`***.
La correzione per cui sarà ***`%hd`*** invece di ***`%f`*** per a e ***`%hd`*** invece di ***`%d`*** per b.
In più possiamo correggere la riga ***`scanf("%f", &a)`*** in ***`scanf("%d", &a)`*** per leggere a come un intero anziché come un float.



Analisi del codice , ricerca e correzione degli errori dello stesso.

Andiamo avanti nel menù della funzione ***`dividi ()`*** e guardando l'immagine possiamo notare un errore nel codice viene calcolato il resto della divisione (non il risultato) utilizzando l'operatore `%` invece che l'operatore di divisione`/`. Andiamo quindi a correggere il codice con l'operatore `/` per ottenere il risultato della divisione tra a e b.



Come ultimo errore identificato possiamo vedere in figura nella funzione ***`ins_string(),`*** che quando si applica lo ***`scanf()`*** per leggere una stringa, non è necessario utilizzare l'operatore ***`&`*** per stringa. La correzione che andremo ad effettuare prevede semplicemente scrivere ***`scanf("%s", stringa)`***; senza l'operatore ***`&`***.