**S2-L5**

Per agire come un Hacker bisogna capire come pensare fuori dagli schemi.

L’esercizio di oggi ha lo scopo di allenare l’osservazione critica. Dato il codice in allegato, si richiede allo studente di:

● Capire cosa fa il programma senza eseguirlo.

● Individuare dal codice sorgente le casistiche non standard che il programma non gestisce (esempio, comportamenti potenziali che non sono stati contemplati).

● Individuare eventuali errori di sintassi / logici.

● Proporre una soluzione per ognuno di essi.

**Spiegazione:** Il programma consiste in un assistente digitale che ci aiuta nello svolgimento di tre operazioni, eseguire una moltiplicazione oppure una divisione, tra due numeri inseriti e ci dà la possibilità di inserire una stringa.

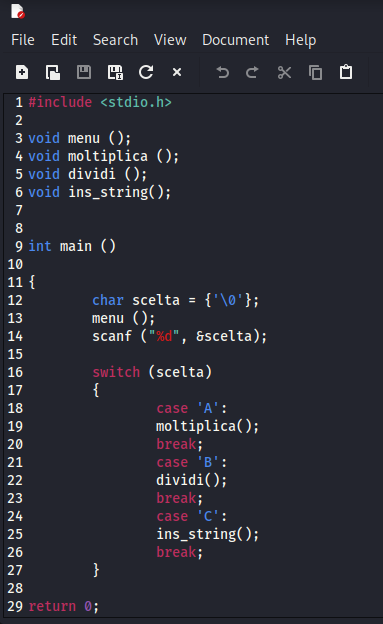
Analizzando il programma, troviamo inclusa la libreria stdio.h che include le funzioni printf e scanf che servono rispettivamente per stampare sullo schermo ciò che contiene(printf) e per immettere dati nel programma (scarf). In seguito, troviamo dichiarate tutte le funzioni di cui avrà bisogno il programma (void menu (), void moltiplica (), void dividi (), void ins\_string()), vengono dichiarate con il void perchè non hanno bisogno di dati in ingresso e non daranno dati in uscita, semplicemente verrà eseguita un'operazione al loro interno. La prima funzione che troviamo è il Main ovvero dove inizia e dove viene effettivamente eseguito il programma, all’interno troviamo tutte le funzioni dichiarate in precedenza, che vengono richiamate in base alle necessità. La prima cosa che notiamo all’interno è il richiamo del menu che ci servirà per decidere quale operazione vogliamo effettuare, con la riga di codice: “ scanf ("%d", &scelta);” andiamo ad immettere quale sarà effettivamente la nostra scelta che appunto verrà salvata nella suddetta variabile (il simbolo “%” serve per indicare dove inserire l’argomento mentre la lettera successiva specifica il tipo di dato, in questo caso d sta per intero. Il simbolo “&” serve per assegnare effettivamente il valore alla variabile), la nostra scelta verrà controllata da uno switch che paragonerà il carattere da noi immesso con i casi presenti al suo interno, se ci saranno corrispondenze, verranno eseguite le operazioni presenti.

La funzione void menu () ha al suo interno la semplice presentazione dell’assistente virtuale, che ci verrà mostrata a schermo grazie alla funzione printf.

La funzione void moltiplica () come dice il nome ci servirà per eseguire la moltiplicazione. Al suo interno troviamo la dichiarazione di due variabili short int che andremo poi ad inserire con la funzione scanf , la dichiarazione della variabile prodotto che effettuerà l’operazione e salverà al suo interno il risultato e il printf finale che ci servirà per visualizzare i numeri utilizzati per l’operazione ed il loro prodotto.

La funzione void divivi() presenta al suo interno due variabili di tipo int che ci serviranno per contenere i numeri che andremo ad utilizzare per l’operazione, anche qui utilizzeremo la funzione scanf per l’inserimento. La variabile int divisione effettuerà l’operazione di divisione e terrà salvato al suo interno il risultato che andremo poi a visualizzare con il print finale.

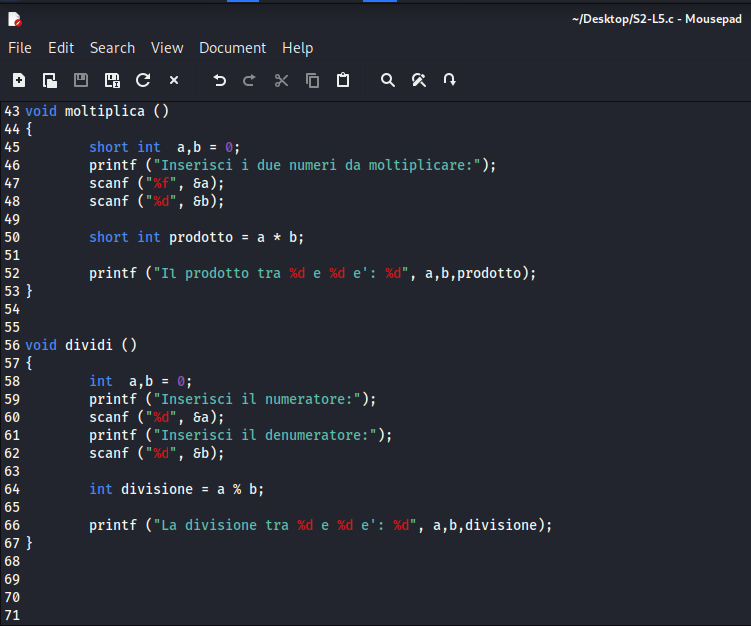
La funzione void ins\_string() servirà per l’inserimento della stringa che verrà salvata all’interno dell’array stringa (char stringa [10]) che come indicato può contenere al suo interno 10 elementi che vanno da 0 per la prima posizione a 9 per l’ultima.



**Osservazione: (riga 12)** Nella funzione Main alla variabile scelta, viene assegnato un valore nullo, cosa di cui non abbiamo bisogno, in quanto avviene automaticamente quando la variabile viene dichiarata ( char scelta);

**Errore 1: (riga 14)** Nella funzione Main quando si richiede l'inserimento della scelta, viene utilizzato %d che si utilizza per i numeri interi, invece di %c che deve essere utilizzato in questo caso, inquanto la variabile è di tipo char;

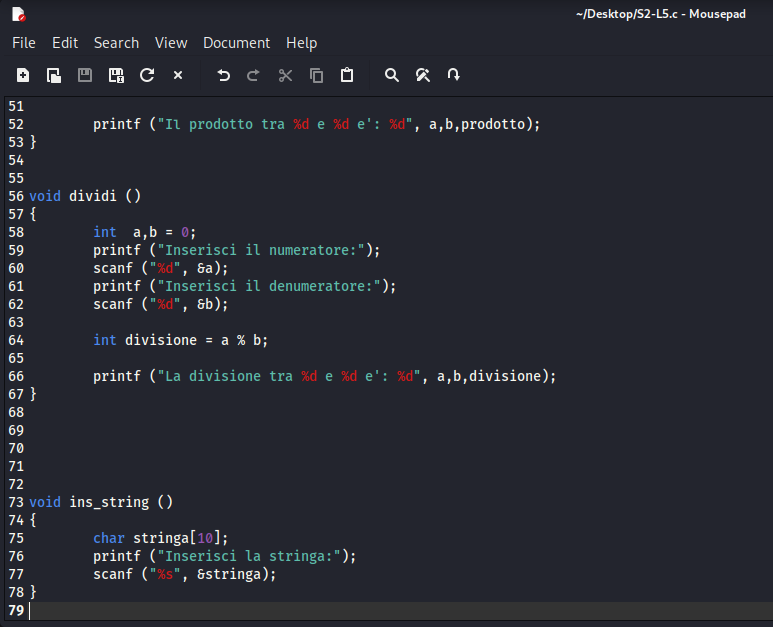
**Errore 2: (riga 16 - 27)** Nella funzione Main avendo uno switch che ci controlla la scelta inserita, dobbiamo aggiungere il default, ovvero, dove andremmo a finire nel caso in cui il carattere inserito non sia presente in nessun case specificato nello switch;



**Errore 3: (riga 47 - 52)** Nella funzione Moltiplica, viene utilizzata %f che si utilizza per i float, mentre noi abbiamo bisogno di %hd avendo in immissione una variabile di tipo short int (che è un int ma più piccolo, per la precisione una variabile int va Da -2.147.483.648 a 2.147.483.647 mentre una variabile short Da -32.768 a 32.767), andiamo così a sostituire anche i %d con i %hd.

**Osservazione: (riga 58-67)** Nella funzione Dividi, troviamo le variabili di tipo intero, nel caso in cui dovremmo aver bisogno di risultati più precisi che contengano la virgola, potremmo utilizzare delle variabili float, naturalmente sostituendo anche tutti i %d con %f;

**Errore 4: (riga 64)** Nella funzione Dividi, viene utilizzato il simbolo % che serve per il modulo, ovvero per sapere il resto della divisione, nel nostro caso dovendo eseguire una semplice divisione abbiamo bisogno del simbolo /.



**Errore 5: (riga 77)** Nella funzione ins\_string quando facciamo effettivamente inserire la frase all'utente, inseriamo il numero 9 in modo da far leggere soltanto 9 caratteri, così da evitare l'overflow(%9s), poi è presente il simbolo & che viene utilizzato per mettere effettivamente il carattere scritto all'interno della variabile, ma in questo caso essendo appunto un array ovvero un puntatore, non ne abbiamo bisogno.

**CODICE CORRETTO**

