# PROGETTO S2 – BUG HUNTING

---

**Traccia:**

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

Esercizio_10_Epicode.c

2

Il programma è un assistente digitale che ci aiuta ad effettuare 3 tipi di operazioni: divisione di due numeri, moltiplicazione di due numeri e inserimento di una stringa.

Nel caso delle due operazioni matematiche, l'utente potrà inserire manualmente i due numeri da dividere o moltiplicare e il programma restituirà il risultato.

Questo codice non prevede alcune casistiche o comportamenti:

- Il programma non mette un limite al numero di caratteri che possono essere inseriti dall'utente. Ad esempio, nella scelta dell'operazione o nella scelta dei numeri da moltiplicare o dividere, in cui l'utente potrebbe inserire un numero superiore a quello assimilabile dalla variabile di tipo char, ovvero da 0 a 255. Se l'utente inserisse il numero 256, il programma potrebbe crashare.

Inserirei un limitatore di caratteri e renderei i numeri in input e in output di tipo float, cambiando anche gli indicatori del tipo di variabile in %f.

Nel programma vi sono anche degli errori di sintassi e logici:

- Nello statement switch della funzione “int main” è sbagliata la sintassi e dovrebbere essere riscritta nel seguente modo:
  {
  case 1 <moltiplica()> [break];
  case 2 <dividi()> [break];
  case 3 <ins_string()> [break];
  }
- Le funzioni menu, moltiplica e dividi, non dovrebbero essere di tipo “void”, dato che in questo modo non possono produrre un output. Sostituirei il tipo di variabile con “int”.
- Nelle funzioni moltiplica e dividi inizializzerei anche a 0 anche le variabili “prodotto” e “divisione”.
- Nella funzione moltiplica, per la variabile prodotto sono indicati due tipi di variabile, short e int.
- Nella funzione dividi, alla variabile divisione, l’operazione di divisione restituisce il resto della divisione perché l’operatore è “%”. Per ottenere il risultato della divisione l’operatore giusto da utilizzare è “/”. Inoltre all’interno di questa funzione, inserirei un costrutto if-else che mi permette di dare in output il risultato della divisione solo nel caso il denominatore sia diverso da 0, mentre se il denominatore inserito sarà zero, l’output sarà la stringa “Non si può dividere un numero per 0”