

# Bug Hunting - Prima Parte

## 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

## Capire cosa fa il programma senza eseguirlo

Il programma si presenta come un assistente digitale pronto ad aiutare l'utente a svolgere dei compiti.

Relativamente a questi inizialmente mostra un menu tramite la funzione **"menu"** proponendogli di fare una scelta tra:

- A) Moltiplicare due numeri
- B) Dividere due numeri
- C) Inserire una stringa

Dopo aver letto la scelta dell'utente esegue la funzione corrispondente:

**"moltiplica"** (per la moltiplicazione)

- Chiede all'utente di inserire due numeri.
- Calcola e visualizza il prodotto di questi numeri.

**"dividi"** (per la divisione)

- Chiede all'utente di inserire due numeri (un numeratore e un denominatore).
- Calcola e mostra il risultato della loro divisione (intesa come divisione intera).

**"ins\_string"** (per l'inserimento di una stringa)

- Chiede all'utente di inserire una stringa e la memorizza. La lunghezza massima della stringa è di 9 caratteri (più il carattere nullo di terminazione).

## Individuare le casistiche non standard che il programma non gestisce

- Il programma non considera la possibilità che l'utente inserisca le scelte (A, B, C) in minuscolo (a, b, c). Nemmeno che inserisca un carattere che non sia A,B,C, oppure un numero oppure una combinazione di lettere e numeri.
- Il programma non effettua alcuna verifica che gli input numerici per la moltiplicazione e la divisione siano effettivamente numeri interi.
- Il programma non effettua controlli sulla natura della stringa inserita. L'utente potrebbe inserire caratteri speciali, numeri, o una stringa vuota.
- Il programma non effettua alcuna verifica sulla grandezza dei numeri inseriti. Specialmente con **"short int"** potrei avere dei problemi con numeri molto grandi.

- Il programma non gestisce casi come la divisione per zero in “dividi” o inserimento di valori non validi in “moltiplica”.
- Non c'è un ciclo per consentire all'utente di riprovare in caso di errore nell'inserimento delle scelte nel menu.
- Il programma termina dopo un'operazione invece di offrire all'utente la possibilità di eseguire altre operazioni o di chiudere il programma.
- Nonostante la lunghezza dell'input sia limitata a 10 caratteri, non c'è una validazione esplicita o un messaggio di errore se l'utente tenta di inserire una stringa più lunga.

## Individuare eventuali errori di sintassi / logici e proporre una soluzione per ognuno di essi

Noto che nella funzione **main** usa %d per leggere char scelta.

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

Soluzione: Sostituire %d con %c in scanf e aggiungere uno spazio per ignorare qualsiasi carattere newline precedente

Nella funzione **moltiplica** a e b non sono inizializzate correttamente. Inoltre usa %f per leggere short int.

```
short int a,b = 0;
printf ("Inserisci i due numeri da moltiplicare:");
scanf ("%f", &a);
scanf ("%d", &b);

short int prodotto = a * b;

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

Soluzione: Inizializzare a e b separatamente e usare %hd in scanf affinché il programma legga correttamente i numeri interi brevi forniti dall'utente.

Nella funzione **dividi** usa l'operatore % che è il modulo, non la divisione.

```
int divisione = a % b;
```

Soluzione: sostituire % con /

Nella funzione ins\_string usa &stringa nonostante stringa sia già un indirizzo che punta all'inizio dell'array di caratteri. Non è corretto usare l'operatore di indirizzo &.

```
char stringa[10];
printf ("Inserisci la stringa:");
scanf ("%s", &stringa);
```

Soluzione: sostituire **scanf ("%s", &stringa)** con **scanf ("%s", stringa)**

## Compito corretto senza aggiunta di miglirie per problematiche non standard:

```
#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':
            moltiplica();
            break;
        case 'B':
            dividi();
            break;
        case 'C':
            ins_string();
            break;
    }

    return 0;
}

void menu() {
    printf("Benvenuto, sono un assistente 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() {
    short int a = 0, b = 0;
    printf("Inserisci i due numeri da moltiplicare:");
    scanf("%hd", &a);
    scanf("%hd", &b);

    short int prodotto = a * b;

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

void dividi() {
    int a = 0, b = 0;
    printf("Inserisci il numeratore:");
    scanf("%d", &a);
    printf("Inserisci il denominatore:");
    scanf("%d", &b);

    int divisione = a / b;

    printf("La divisione tra %d e %d e': %d\n", a, b, divisione);
}

void ins_string() {
    char stringa[10];
    printf("Inserisci la stringa:");
    scanf("%s", stringa);
}
```