



Compito di Raffaele Iacono.

## **Esercizio sul Bug Hunting**

**Iniziamo con la nostra avventura esponendo le prime due problematiche del codice fornito.**

**Si tratta di errori grossolani e facilmente individuabili.**

**int main ()**

**Porzione di codice con problematiche**

**{**

**char scelta = {'\0'};**

**menu ();**

```
scanf ("%d", &scelta);
```

```
switch (scelta)
```

```
{
```

```
    case 'A':
```

```
        moltiplica();
```

```
        break;
```

```
    case 'B':
```

```
        dividi();
```

```
        break;
```

```
    case 'C':
```

```
        ins_string();
```

```
        break;
```

```
}
```

```
return 0;
```

```
}
```

## Porzione di codice Fixata

```
int main()
{
    char scelta = '\0'; // Usare '\0' invece di {'\0'}
    menu();

    scanf(" %c", &scelta); // Fixato con %c per la
lettura di un carattere


    switch (scelta)
    {
        case 'A':
            moltiplica();
            break;

        case 'B':
            dividi();
            break;

        case 'C':
            ins_string();
```

```
break;  
}
```

## Descrizione delle problematiche

All'interno della variabile di tipo Char c'e' un chiaro errore di sintassi (suppongo volutamente evidente), nello scanf invece troviamo un errore di tipo logico, il % in C indica come deve essere letto un numero ed in questo caso l'esercizio buggato presentava nello scanf un %d che si riferisce a tipologie di numeri interi.

## Ottimizzazione per esecuzione su file.exe

**sleep(5);**// Per aumentare i secondi di visibilita' dell'output prodotto ho aggiunto questa funziona fornita dalla libreria <unistd.h>

```
return 0;
```

Questa funzione ci permette di aspettare 5 secondi dopo che il programma ha esplicato la sua funzione per implementarla bisogna aggiungere la libreria <unistd.h>.

# Bug nelle funzioni

## Porzione di Codice Buggato

```
void moltiplica ()  
{  
  
    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);  
  
}
```

**void dividi ()**

**{**

**int a,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",  
a,b,divisione);**

**}**

**void ins\_string ()**

**{**

**char stringa[10];**

**printf ("Inserisci la stringa:");**

**scanf ("%s", &stringa);**

```
}
```

## **Porzione di codice Fixata**

```
void moltiplica()
```

```
{
```

```
    int a, b; // Cambiato il tipo da short int a int per  
    permettere la moltiplicazione corretta
```

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

```
    scanf("%d %d", &a, &b); // Usato %d per la  
    lettura di interi
```

```
    int prodotto = a * b;
```

```
    printf("Il prodotto tra %d e %d è: %d\n", a, b,  
    prodotto); // Aggiunto uno/n
```

```
}
```

```
void dividi()
```

```
{
```

```
int a, b;  
printf("Inserisci il numeratore:");  
scanf("%d", &a);  
printf("Inserisci il denominatore:");  
scanf("%d", &b);  
  
if (b != 0) // Aggiunto il controllo per evitare la  
divisione per zero  
{  
  
int divisione = a / b; // Usato / per la divisione  
corretta  
  
printf("La divisione tra %d e %d è: %d\n", a,  
b, divisione); // Aggiunto uno/n  
}  
  
else  
  
{  
  
printf("Impossibile dividere per zero\n"); //  
Messaggio di errore per divisione per zero
```



```

    }
}

void ins_string()
{
    char stringa[100]; // Aumentata la dimensione
    dell'array

    printf("Inserisci la stringa:");

    scanf("%s", stringa); // Non è necessario
    l'operatore & per gli array di caratteri

    printf("Hai inserito la stringa: %s\n", stringa); //
    Aggiunto uno/n
}

```

## Descrizione

I primi due errori si trovano all'interno della **funzione `moltiplica()`**.

Il primo errore e' nella variabile, una variabile short int non permette una moltiplicazione corretta e per questo motivo e' stata cambiata con una variabile di tipo int; nello scanf della medesima funzione vi erano

**un %f ed un %d ed e' stato fixato cambiando il %f col %d.**

**Nella **funzione divisione** e' stato aggiunto un sistema di sicurezza che non permettesse la divisione per 0 e all'interno della variabile e' stato sostituito l'operatore di resto con quello della divisione.**

**Nella funzione **inserimento stringa** e' stata aumentata la dimensione dell' array e rimosso l'operatore logico & all'interno dello scanf poiche' superfluo.**