

S2/L5

BUG * HUNTING

INTRODUZIONE

In questo progetto si analizzerà il codice in C per trovare degli errori e bug e si proporranno soluzioni per creare un codice funzionante

CODICE:

```
#include <stdio.h>
```

```
void menu ();
```

```
void moltiplica ();
```

```
void dividi ();
```

```
void ins_string();
```

```
int main ()
```

```
{  
    char scelta = {'\0'}; dovrebbe essere char scelta = '\0'
```

```
    menu ();
```

```
    scanf ("%d", &scelta); scanf ("%c", &scelta) perché legge e memorizza un carattere
```

```
    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,b = 0; int a,b;
    printf ("Inserisci i due numeri da moltiplicare:");
    scanf ("%f", &a); scanf ("%d", &a) Il formato %f è utilizzato per leggere numeri in virgola
    mobile, non numeri interi.
    scanf ("%d", &b);

    short int prodotto = a * b; int = a * b;

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

void dividi ()

```

```

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

    int divisione = a % b; float divisione = (float)a / b per la divisione;
    printf ("La divisione tra %d e %d e': %d", a,b,divisione); printf ("La divisione tra %d e %d è:
%.2f\n", a , b, divisione);

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

```

SPIEGAZIONE CORREZIONE ERRORI

Codice Corretto

```

#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;
        default:
            printf("Scelta non valida.\n");
            break;
    }

    return 0;
}

void menu()
{
    printf("Benvenuto, sono Eleonora la tua fantastica assistente digitale, posso aiutarti a sbrigare alcuni compiti meglio di te\n");
    printf("Come posso aiutarti dopo il caffè?\n");
    printf("A >> Moltiplicare due numeri\nB >> Dividere due numeri\nC >> Inserire una stringa\n");
}

void moltiplica()
{
    int a, b;
    printf("Inserisci i due numeri da moltiplicare:\n");
    scanf("%d", &a);
    scanf("%d", &b);

    int prodotto = a * b;

```

```

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

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

    if (b != 0)
    {
        float divisione = (float)a / b;
        printf("La divisione tra %d e %d è: %.2f\n", a, b, divisione);
    }
    else
    {
        printf("Errore: divisione per zero.\n");
    }
}

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

```

screen eseguito dal mio Notion.so <https://www.notion.so/S2-La5ab6bcd70544c53a28226071976243b?pvs=4>

1. Tipo variabile errato:

Errore: la variabile scelta è correttamente dichiarata come char in quanto rappresenta un singolo carattere. Invece, quando il valore viene preso in input dall'utente, si utilizza %d come specificatore di formato per gli interi al posto di %c che è usato per i caratteri.

Soluzione: per leggere correttamente il carattere inserito dall'utente, è necessario modificare la chiamata a scanf e utilizzare %c. Ho aggiunto uno spazio prima del simbolo %c, per evitare di considerare eventuali spazi bianchi o newline nel buffer.

2. Moltiplicazione:

Errore: quando si leggono i numeri per la moltiplicazione, viene utilizzato %d come formato anziché %f che è specifico per numeri in virgola mobile.

Soluzione: si utilizza %d per acquisire un numero intero.

3. Divisione:

Errore: nella funzione `dividi()`, l'operatore `%` viene utilizzato per calcolare il risultato della divisione tra due numeri interi, quest'operatore in realtà viene utilizzato per calcolare il resto della divisione.

Soluzione: per ottenere il risultato corretto della divisione tra due numeri, si utilizza l'operatore `/` anziché `%`.

4. Gestione scorretta delle stringhe in `ins_string()`:

Errore: non è corretto utilizzare `&stringa` in `scanf` perché `stringa` è già un puntatore ad un array.

Soluzione: modificare la chiamata a `scanf` rimuovendo il segno `&`.