

# PROVA PRATICA S2/L5

## “Bug Hunting”

Foto programma iniziale (con bug):

```
GNU nano 7.2
#include <stdio.h>

void menu ();
void moltiplica ();
void dividi ();
void ins_string();

int main ()
{
    char scelta = {'\0'};
    menu ();
    scanf ("%d", &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,b = 0;
    printf ("Inserisci i due numeri da moltiplicare:");
    scanf ("%f", &a);
    scanf ("%d", &b);
```

```
GNU nano 7.2

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;
    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);
}
```

## Descrizione del programma:

Il programma vuole riprodurre una sorta di assistente digitale che permette di svolgere alcune opzioni come:

- **Moltiplicare due numeri**
- **Dividere due numeri**
- **Inserire una stringa**

In ogni opzione il programma chiede all'utente di inserire un valore come "input" (funzione "*scanf*") per poi restituirgli come "output" (funzione "*printf*") il risultato dell'operazione selezionata: come ad esempio il prodotto dei due numeri nell'opzione della moltiplicazione o il quoziente nel caso della divisione.

Il programma quindi si svilupperà in questo modo:

1. mostrerà all'utente un menù con queste opzioni (moltiplicare, dividere o inserire una stringa) e attenderà l'input dell'utente.
2. In base alla scelta dell'utente (A, B o C), il programma eseguirà la funzione corrispondente.
3. Una volta eseguita l'operazione scelta dall'utente, il programma terminerà l'esecuzione.

## Correzione delle criticità

Di seguito allego il programma completo corretto e analizzo singolarmente tutte le casistiche non gestite insieme agli eventuali errori riscontrati.

```
GNU nano 7.2
#include <stdio.h>

//inserimento delle funzioni utilizzate
void menu();
void moltiplica();
void dividi();
void ins_string();

//funzionamento del programma
int main() {
    char scelta;

    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");
    }

    return 0;
}

//definizione della funzione 'menu'
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");
}

//definizione della funzione 'moltiplica'
void moltiplica() {
    double a, b;
    printf("Inserisci i due numeri da moltiplicare: ");
    scanf("%lf %lf", &a, &b);

    double prodotto = a * b;

    printf("Il prodotto tra %.2lf e %.2lf è: %.2lf\n", a, b, prodotto);
}
```

```
GNU nano 7.2
}

return 0;
}

//definizione della funzione 'menu'
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");
}

//definizione della funzione 'moltiplica'
void moltiplica() {
    double a, b;
    printf("Inserisci i due numeri da moltiplicare: ");
    scanf("%lf %lf", &a, &b);

    double prodotto = a * b;

    printf("Il prodotto tra %.2lf e %.2lf è: %.2lf\n", a, b, prodotto);
}

//definizione della funzione 'dividi'
void dividi() {
    double a, b;
    printf("Inserisci il dividendo: ");
    scanf("%lf", &a);
    printf("Inserisci il divisore: ");
    scanf("%lf", &b);

    if (b != 0) {
        double divisione = a / b;
        printf("La divisione tra %.2lf e %.2lf è: %.2lf\n", a, b, divisione);
    } else {
        printf("Impossibile dividere per zero.\n");
    }
}

//definizione della funzione 'ins_string'
void ins_string() {
    char stringa[150];
    printf("Inserisci la stringa: ");
    scanf("%s", stringa);
    printf("Hai inserito: %s\n", stringa);
}
```

## Prima correzione:

Nella funzione principale '*main ()*' è stata inserita una funzione condizionale '*switch*', ma all'interno delle opzioni non è stato inserito lo scenario di '*default*' che si verificherebbe quando l'utente seleziona una scelta diversa da "A, B e C".

Di conseguenza non sono state gestite tutte le casistiche possibili.

Per correggere questa mancanza si è provveduto inserendo un'opzione di '*default*'.

- **Codice iniziale:**

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

    switch (scelta)
    {
        case 'A':
            moltiplica();
            break;
        case 'B':
            dividi();
            break;
        case 'C':
            ins_string();
            break;
    }

    return 0;
}
```

- **Codice corretto:**

```
//funzionamento del programma
int main() {
    char scelta;

    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");
    }

    return 0;
}
```

## Seconda correzione:

Nella funzione *'moltiplica ()'*, le variabili "a" e "b" sono state dichiarate come variabili di tipo *'short int'*, ma nella prima funzione *'scanf'* è stato utilizzato *'%f'* per ricevere in 'input' dall'utente un *'INT'* ma ciò non è corretto, si sarebbe dovuto utilizzare *'%d'*.

Inoltre il fatto che le variabili siano state dichiarate come di tipo *'short int'* non è funzionale.

Infatti il range dei numeri interi che possono essere dichiarati come *'short int'* va da [-32.768 to +32.767] ma c'è la concreta possibilità che il prodotto possa essere maggiore, considerando inoltre che non vengono considerati numeri reali e ciò molto limitante.

Per correggere questi errori si è provveduto a dichiarare tutte le variabili come *'double'* così da evitare qualsiasi limitazione, conseguentemente nelle funzioni *'scanf'* per ricevere l'input dall'utente si è utilizzato *'%lf'*.

- **Codice iniziale:**

```
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);
}
```

- **Codice corretto:**

```
//definizione della funzione 'moltiplica'
void moltiplica() {
    double a, b;
    printf("Inserisci i due numeri da moltiplicare: ");
    scanf("%lf %lf", &a, &b);

    double prodotto = a * b;

    printf("Il prodotto tra %.2lf e %.2lf è: %.2lf\n", a, b, prodotto);
}
```

### Terza correzione:

Nella funzione `'dividi()'` non è stato considerato lo scenario in cui l'utente divide un numero per 0. La divisione per 0 è un'operazione delicata e la mancata gestione di tale operazione può portare allo sviluppo di errori a livello di istruzioni.

Inoltre nella funzione `'int divisione'` è stato utilizzato impropriamente il simbolo `'%'` al posto del simbolo della divisione `'/'`.

Infine le variabili nella funzione `'dividi()'` sono state dichiarate come `'int'` il che oltre a permettere di inserire solo un range di valori interi [-2.147.483.648 to 2.147.483.647] non permette l'inserimento, sia in `'input'` che in `'output'`, di valori non interi (numeri con la virgola) e ciò vorrebbe dire approssimare il quoziente in maniera eccessiva.

Per risolvere questi problemi è stata inizialmente inserita una funzione `'if -else'` per gestire il caso in cui si divida un numero per 0, successivamente sono state modificate tutte le variabili andando ad esplicitarle come `'double'`.

- **Codice iniziale:**

```
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);
}
```

- **Codice corretto:**

```
//definizione della funzione 'dividi'
void dividi() {
    double a, b;
    printf("Inserisci il dividendo: ");
    scanf("%lf", &a);
    printf("Inserisci il divisore: ");
    scanf("%lf", &b);

    if (b != 0) {
        double divisione = a / b;
        printf("La divisione tra %.2lf e %.2lf è: %.2lf\n", a, b, divisione);
    } else {
        printf("Impossibile dividere per zero.\n");
    }
}
```

#### Quarta correzione:

Nella funzione 'scanf' della funzione principale 'ins\_string' è stato inserito l'operatore di indirizzo '&' ma il formato specificatore '%s' utilizzato in 'scanf()' per leggere una stringa non lo richiede. Inoltre nella funzione 'char stringa' è stato inserito un numero massimo di 10 caratteri, questo rende il programma vulnerabile perché se l'utente inserirebbe in 'input' un valore superiore di caratteri genererebbe un **overflow** che potrebbe essere utilizzato da un eventuale attaccante per prendere il "controllo" del programma.

Per risolvere queste problematiche è stato rimosso l'operatore di indirizzo '&' ed è stato ampliato il numero di caratteri possibili a 150.

- **Codice iniziale:**

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

- **Codice corretto:**

```
//definizione della funzione 'ins_strign'  
void ins_string() {  
    char stringa[150];  
    printf("Inserisci la stringa: ");  
    scanf("%s", stringa);  
    printf("Hai inserito: %s\n", stringa);  
}
```



## Test programma corretto:



```
File Actions Edit View Help
zsh: corrupt history file /home/kali/.zsh_history
(kali@kali)~$ cd Desktop

(kali@kali)~/Desktop$ sudo ./esercizio_S2L5
[sudo] password for kali:
Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti.
Come posso aiutarti?
A >> Moltiplicare due numeri
B >> Dividere due numeri
C >> Inserire una stringa
A
Inserisci i due numeri da moltiplicare: 2
5
Il prodotto tra 2.00 e 5.00 è: 10.00

(kali@kali)~/Desktop$ sudo ./esercizio_S2L5
Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti.
Come posso aiutarti?
A >> Moltiplicare due numeri
B >> Dividere due numeri
C >> Inserire una stringa
B
Inserisci il dividendo: 5
Inserisci il divisore: 2
La divisione tra 5.00 e 2.00 è: 2.50

(kali@kali)~/Desktop$ sudo ./esercizio_S2L5
Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti.
Come posso aiutarti?
A >> Moltiplicare due numeri
B >> Dividere due numeri
C >> Inserire una stringa
C
Inserisci la stringa: Ciao Manu!
Hai inserito: Ciao
```