

1) Descrizione del Funzionamento

Il programma si propone come un assistente digitale che può eseguire tre operazioni:

1. Moltiplicare due numeri (opzione 'A').
2. Dividere due numeri (opzione 'B').
3. Inserire una stringa (opzione 'C').

L'utente viene accolto con un menu e deve scegliere una delle tre opzioni. In base alla scelta dell'utente, il programma chiama la funzione corrispondente.

2) Analisi delle Problematiche

1. ****Errore di Sintassi nella Scelta del Menu**:**

- ****Errore**:** La variabile ``scelta`` viene letta come ``char``, ma viene usata con ``%d`` in ``scanf``.
- ****Soluzione**:** Cambiare ``%d`` in ``%c`` in ``scanf`` per leggere correttamente il carattere.

2. ****Errore di Tipo in ``moltiplica``**:**

- ****Errore**:** Viene utilizzato ``%f`` per leggere una variabile di tipo ``short int``.
- ****Soluzione**:** Usare ``%hd`` per leggere ``short int``.

3. ****Errore di Tipo in ``dividi``**:**

- ****Errore**:** La divisione viene effettuata usando l'operatore ``%`` invece di ``/`` per ottenere il risultato corretto della divisione.
- ****Soluzione**:** Usare l'operatore ``/`` per la divisione.

4. ****Buffer Overflow in ``ins_string``**:**

- ****Errore**:** Non viene specificata la lunghezza massima della stringa in ``scanf``, causando potenzialmente un buffer overflow.
- ****Soluzione**:** Specificare la lunghezza massima del buffer in ``scanf``.

5. ****Non Gestione dei Numeri Zero nella Divisione**:**

- ****Errore**:** Il programma non gestisce il caso in cui il denominatore sia zero.

- **Soluzione**: Aggiungere un controllo per evitare la divisione per zero.

6. **Inizializzazione Errata delle Variabili**:

- **Errore**: In `moltiplica` e `dividi`, solo una delle variabili viene inizializzata.

- **Soluzione**: Inizializzare entrambe le variabili correttamente.

3) Ecco il codice corretto con le modifiche suggerite:

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

```
void menu();
```

```
void moltiplica();
```

```
void dividi();
```

```
void ins_string();
```

```
int main()
```

```
{
```

```
    char scelta = '\0';
```

```
    menu();
```

```
    scanf(" %c", &scelta); // Usare %c per leggere il carattere
```

```
    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 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 %hd e %hd e': %hd\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);

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

```

```

void ins_string()
{
    char stringa[10];
    printf("Inserisci la stringa:");
    scanf("%9s", stringa); // Limitare la lunghezza della stringa a 9 caratteri

    printf("Hai inserito: %s\n", stringa);
}

```

4) Spiegazione delle Modifiche

1. ****Lettura della Scelta del Menu**:**

- Cambiato `%d` in `%c` per la lettura della variabile `scelta`.

2. ****Lettura delle Variabili in `moltiplica`**:**

- Utilizzato `%hd` per leggere `short int`.

3. ****Correzione della Divisione in `dividi`**:**

- Cambiato l'operatore `%` in `/` e aggiunto il controllo per la divisione per zero.

4. ****Limitazione della Lunghezza della Stringa in `ins_string`**:**

- Utilizzato `%9s` per evitare buffer overflow.