

ESERCIZIO 03.02.2023

COSA FA IL PROGRAMMA SENZA ESEGUIRLO?

Il programma è un assistente digitale, in grado di A: Moltiplicare due numeri, B: Dividere due numeri e C : Inserire una stringa.

In base alla scelta dell'utente, viene eseguita una delle tre funzioni.

La funzione "moltiplica()" chiede all'utente di inserire due numeri e poi calcola il risultato. La funzione "dividi()" chiede all'utente di inserire il numeratore e il denominatore e poi calcola e visualizza il loro resto.

La funzione "ins_string()" chiede all'utente di inserire una stringa di caratteri.

CASISTICHE NON STANDARD

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

```
switch (scelta)
```

```
{
```

```
case 'A':
```

```
    moltiplica();
```

```
    break;
```

```
case 'B':
```

```
    dividi();
```

```
    break;
```

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

```
    break;
```

```
}
```

Ci sono due casistiche non standard nella prima parte di codice:

scanf, con il formato "%d", perchè la scelta è un carattere e non un intero. Si dovrebbe utilizzare "%c".

la variabile scelta viene dichiarata come carattere, ma viene confrontata con costanti di tipo intero. Si dovrebbe utilizzare costanti di tipo carattere.

```

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);
}

```

In questa parte di codice c'è una casistica non standard ossia l'uso scorretto di scanf con il formato “%f” per leggere il primo numero, da errore perché la variabile “a” è dichiarata come short int che è un tipo intero a 16 bit. Si dovrebbe utilizzare "%hd"

```

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

In questa ultima parte di codice:

l'utilizzo di `scanf` con il formato `"%s"` per leggere una stringa può causare un buffer overflow se la stringa inserita è più lunga della dimensione massima del buffer.

ERRORI SINTATTICI E LOGICI E SOLUZIONE PER OGNUNO DI ESSI

Questo codice presenta alcuni errori sintattici e logici.

La funzione **`scanf`** in **`main`** usa **`%d`** per leggere una variabile di tipo `char`, ma dovrebbe usare **`%c`**.

La funzione **`moltiplica`** usa **`%f`** per leggere due variabili di tipo **`short int`**, che dovrebbe essere **`%hd`**.

La funzione **`dividi`** utilizza la modulo (**`%`**) invece della divisione (**`/`**).

La funzione **`ins_string`** passa l'indirizzo della stringa al posto del nome dell'array, dovrebbe usare **`%s`** invece di **`&stringa`**.