

È un programma utile che propone un assistente digitale in grado di aiutare a risolvere dei compiti. Nello specifico dopo la possibilità di scelta tramite un menù bisogna decidere se si ha bisogno di aiuto nel risolvere una moltiplicazione, una divisione o inserire una stringa e in base alla propria decisione il programma fornirà i risultati.

Nel seguente codice ho aggiunto dei commenti per correggere i comportamenti potenziali non contemplati, gli errori logici e di sintassi:

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

```
void menu ();
```

```
void moltiplica ();
```

```
void dividi ();
```

```
void ins_string();
```

```
int main ()
```

```
{
```

```
    char scelta = {'\0'}; /* è un errore utilizzare le graffe perchè in questo modo identifica il char  
    come un array e non come carattere singolo*/
```

```
    menu ();
```

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

```
    switch (scelta)
```

```
    {
```

```
    case 'A':
```

```
        moltiplica();
```

```
        break;
```

```
    case 'B':
```

```
        dividi();
```

```
        break;
```

```
    case 'C':
```

```
        ins_string();
```

```
        break;
```

```
/* Meglio aggiungere un default per coprire le casistiche che il sistema non gestisce e per migliorare la così detta “user experience”*/
```

Default:

```
printf("Non posso elaborare la tua richiesta! Ritenta!");
```

```
break;
```

```
return 404;
```

```
}
```

```
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); /* correggere il tipo dell'argomento dato che il placeholder dello short int è %hd */
```

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

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

```
printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto); /* ripetizione di %d non funzionerebbe*/  
}
```

```
void dividi ()
```

```
{
```

```
    int a,b = 0; /* qui è sbagliato scrivere che sia = 0 perchè sarebbe un errore nel caso il  
    dividendo fosse 0 di conseguenza sarà meglio scrivere >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]; /* in questo caso si ha un rischio di overflow bisognerebbe utilizzare un'altra  
funzione al posto di scanf che si chiama fgets per prendere solo i primi 10 non di più, è importante  
sapere riconoscere l'overflow perchè è una porta aperta per essere hackerata e per far buttare giù  
il sistema*/
```

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

```
    scanf ("%s", &stringa); /* In questo caso la & è sbagliata perchè è come se gli stessi dicendo  
nello stesso puntatore mi devi scrivere tutti i valori della stringa che in realtà occupano banchi di  
memoria sequenziali ma diversi*/
```

```
}
```