

```

1#include <stdio.h>
2void menu ();
3void moltiplica ();
4void dividi ();
5void ins_string();
6
7int main ()
8{
9    char scelta = {'\0'};
10    menu ();
11    scanf ("%d", &scelta);
12
13    switch (scelta)
14    {
15        case 'A':
16            moltiplica();
17            break;
18        case 'B':
19            dividi();
20            break;
21        case 'C':
22            ins_string();
23            break;
24    }
25
26    return 0;
27 }
28
29
30 void menu ()
31 {
32     printf ("Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti\n");
33     printf ("Come posso aiutarti?\n");
34     printf ("A >> Moltiplicare due numeri\nB >> Dividere due numeri\nC >> Inserire una stringa\n");
35
36 }
37
38 void moltiplica ()
39 {
40     short int a,b = 0;
41     printf ("Inserisci i due numeri da moltiplicare:");
42     scanf ("%f", &a);
43     scanf ("%d", &b);
44
45     short int prodotto = a * b;
46
47     printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto);
48 }
49
50
51 void dividi ()
52 {
53     int a,b = 0;
54     printf ("Inserisci il numeratore:");
55     scanf ("%d", &a);
56     printf ("Inserisci il denominatore:");
57     scanf ("%d", &b);
58
59     int divisione = a % b;
60     printf ("La divisione tra %d e %d e': %d", a,b,divisione);
61 }
62
63 void ins_string ()
64 {
65     char stringa[10];
66     printf ("Inserisci la stringa:");
67     scanf ("%s", &stringa);
68 }
69

```

Il programma in questione è un ASSISTENTE DIGITALE, che offre all'utente tre opzioni per moltiplicare due numeri, dividere due numeri, inserire una stringa.

Come vediamo in seguito però, ci sono alcune problematiche di sintassi e logiche.

1- nella dichiarazione di variabile scelta, ella funzione char scelta, l'utilizzo delle parentesi graffe è errato

```

9 char scelta = {'\0'};

```

La soluzione per questa variabile è:
Perché, le parentesi {}, vengono utilizzate per definire valori predefiniti, in questo caso la variabile char non ha bisogno di utilizzarle, perché il valore è vuoto, quindi possiamo

```

9 char scelta = '\0';

```

2- Vediamo come in riga 11 sia sbagliato l'input che deve essere %c, perché utilizziamo la variabile char, con %d leggiamo un intero e non un carattere.

Come vediamo per la scelta dell'utente usiamo switch, ma non è stato gestito il caso in cui l'ente scelga una cosa diversa da A,B,C, manca il default, che sarebbe un'istruzione, che indica all'utente di aver sbagliato scelta, un modo per gestire gli input non previsti.

```

7 int main ()
8 {
9     char scelta = '\0';
10    menu ();
11    scanf ("%d", &scelta);
12
13    switch (scelta)
14    {
15        case 'A':
16            moltiplica();
17            break;
18        case 'B':
19            dividi();
20            break;
21        case 'C':
22            ins_string();
23            break;
24    }
25
26    return 0;
27 }
28
29

```

2- Soluzione:

- %c variabile char, non %d
- default: "scelta non valida.\n", che sarebbe un'istruzione, che indica all'utente di aver sbagliato scelta, un modo per gestire gli input non previsti.

```
7 int main()
8 {
9     char scelta = '\0';
10    menu ();
11    scanf ("%c", &scelta);
12
13    switch (scelta)
14    {
15        case 'A':
16            multiplica();
17            break;
18        case 'B':
19            dividi();
20            break;
21        case 'C':
22            ins_string();
23            break;
24        default:
25            printf("Scelta non valida.\n");
26            break;
27    }
28
29    return 0;
30
31 }
```

3- Funzione multiplica (), c'è la variabile short int, vediamo come in scanf, sia utilizzato %f, che appunto può causare problemi perché ci darà dei valori dopo la virgola, quindi ci potranno essere degli imprevisti.

```
40 short int a,b = 0;
41 printf ("Inserisci i due numeri da moltiplicare:");
42 scanf ("%f", &a);
43 scanf ("%d", &b);
44
```

La soluzione è quella di definire a=0 e b=0 sostituire %f con %d così da leggere numeri di tipo short int.

Teniamo conto che in questo caso il range degli interi short potrebbe non andare bene, in quanto c'è possibilità che il risultato potrebbe andare ben oltre, quindi si consiglia di utilizzare int.

```
40 int a = 0, b = 0;
41 printf ("Inserisci i due numeri da moltiplicare:");
42 scanf ("%d", &a);
43 scanf ("%d", &b);
44
```

4- Funzione dividi(),

Definiamo meglio a, poiché non essendo definita=0, così siamo sicuri che è assegnata vediamo che al posto del dividi c'è %, poi manca la gestione della divisione per 0, che essendo un caso particolare deve avere istruzioni.

```
51 void dividi ()
52 {
53     int a,b = 0;
54     printf ("Inserisci il numeratore:");
55     scanf ("%d", &a);
56     printf ("Inserisci il denominatore:");
57     scanf ("%d", &b);
58
59     int divisione = a % b;
60     printf ("La divisione tra %d e %d e': %d", a,b,divisione);
61 }
62
```

Soluzione:

- Assegnare a= 0 e b=0
- sostituire % con / così andremo a dividere
- per gestire la divisione per 0 diamo il ciclo if else, in modo tale che se b ≠ 0 la divisione viene eseguita se B=0 ci dirà che non si può fare.

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

    if (b != 0) {
        divisione = a / b;
        printf("La divisione tra %d e %d e': %d\n", a, b, divisione);
    } else {
        printf("Non puoi dividere per zero\n");
    }
}
```

Vediamo che viene utilizzato `ins_string` per leggere una stringa dell'input, questo può causare un problema di overflow, se l'input supera i 10 caratteri, e non c'è modo di controllare questi caratteri appunto accade, ciò che non vogliamo.

```
61 }
62
63 void ins_string ()
64 {
65     char stringa[10];
66     printf ("Inserisci la stringa:");
67     scanf ("%s", &stringa);
68 }
69
```

Soluzione, per risolvere un possibile problema d. Overflow, utilizziamola funzione `fgets`, invece di `scanf`. In questo modo `fgets` legge la stringa in input e la memorizza stringa. Poi limitiamo la dimensione massima della stringa all'interno dell'array stringa che è di 100

```
68 void ins_string ()
69 {
70     char stringa[100]; // Definiamo un array di
71     printf("Inserisci la stringa: ");
72     fgets(stringa, sizeof(stringa), stdin); // U
73 }
74
```