

```

1  #include <stdio.h>
2
3  void menu ();
4  void multiplica ();
5  void dividi ();
6  void ins_string();
7
8
9  int main ()
10 {
11     char scelta = {'\0'};
12     menu ();
13     scanf ("%d", &scelta);
14
15     switch (scelta)
16     {
17         case 'A':
18             multiplica();
19             break;
20         case 'B':
21             dividi();
22             break;
23         case 'C':
24             ins_string();
25             break;
26     }
27
28     // while ((scelta != 'A') && (scelta != 'B') && (scelta != 'C'))
29     // {
30     //     printf("Errore, valore inserito non riconosciuto");
31     //     menu();
32     //     scanf ("%c", &scelta);
33     //     switch (scelta)
34     //     {
35     //         case 'A':
36     //             multiplica();
37     //             break;
38     //         case 'B':
39     //             dividi();
40     //             break;
41     //         case 'C':
42     //             ins_string();
43     //             break;
44     //     }
45     // }
46
47     return 0;
48 }

```

CONSEGNA [SETTIMANA 2 LEZIONE 5]

Nella schermata accanto è rappresentato un codice in C, la cui funzione primaria è quella di permettere all'utente di effettuare 3 operazioni (multiplica, dividi, e inserisci stringa).

[INIZIALIZZAZIONE DELLE FUNZIONI]

Dopo aver inserito la libreria stdio.h, sono inizializzate le funzioni *menu*, *multiplica*, *dividi* e *ins_string*, tutte di tipologia void (ovvero senza un ritorno) e senza argomenti in ingresso.

[FUNZIONE MAIN]

Nella funzione main, è stato inserito il costrutto switch, che permette all'utente di scegliere una delle 3 operazioni da effettuare.

Possiamo notare un errore alla riga 14, all'interno dello scanf, in quanto il **data-type**, essendo un char richiede la dicitura %c.

Nonostante la funzione switch sia stata impostata correttamente, non è stato inserito un **gestore delle possibili eccezioni** (in questo caso un inserimento di un valore diverso da parte dell'utente)

In questo caso il programma ignorerà le istruzioni contenute all'interno dello switch per poi chiudersi.

Una possibile soluzione consiste nel costrutto while, controllando la variabile e richiedendo nuovamente l'input se non dovesse corrispondere a A, B o C.

Un ulteriore "errore" è la **mancata indentazione** delle istruzioni alla riga 19-20.

```

33
34 void menu ()
35 {
36     printf ("Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti\n");
37     printf ("Come posso aiutarti?\n");
38     printf ("A >> Moltiplicare due numeri\nB >> Dividere due numeri\nC >> Inserire una stringa\n");
39
40 }

```

```

41
42
43 void moltiplica ()
44 {
45     short int a,b = 0;
46     printf ("Inserisci i due numeri da moltiplicare:");
47     scanf ("%f", &a); // scanf("%hd", &a);
48     scanf ("%d", &b); // scanf("%hd", &b);
49
50     short int prodotto = a * b;
51
52     printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto);
53 }

```

```

54
55
56 void dividi ()
57 {
58     int a,b = 0;
59     printf ("Inserisci il numeratore:");
60     scanf ("%d", &a);
61     printf ("Inserisci il denominatore:");
62     scanf ("%d", &b);
63
64     int divisione = a % b; // float divisione = a / b;
65
66     printf ("La divisione tra %d e %d e': %d", a,b,divisione);
67 }

```

```

68
69
70
71
72
73 void ins_string ()
74 {
75     char stringa[10];
76     printf ("Inserisci la stringa:");
77     scanf ("%s", &stringa); // fgets(stringa, 10, stdin);
78 }

```

[FUNZIONE MENU]

Questa funzione, richiamata dal main, permette la stampa su schermo del “menu di selezione”.

E’ presente un **errore di battitura** alla *riga 36* alla parola “assistente”.

[FUNZIONE MOLTIPLICA]

La funzione moltiplica permette di calcolare la moltiplicazione tra due numeri inseriti dall’utente.

L’errore presente in questa funzione è costituito dai **data-type errati**. Infatti a, b e il prodotto sono stati inizializzati come *short int*, quindi necessitano nelle istruzioni print() e scanf() della dicitura corretta, ovvero %hd oppure %hi. Anche in questo caso non sono stati contemplati eventuali inserimenti errati da parte dell’utente (un utente potrebbe inserire una variabile float).

Una possibile soluzione consiste nell’effettuare un controllo sulle variabili in ingresso e richiedere nuovamente l’input specificando l’errore all’utente.

[FUNZIONE DIVIDI]

La funzione dividi permette di calcolare il risultato della divisione tra numeratore e denominatore, inseriti dall’utente.

L’errore principale in questa funzione risiede nell’uso dell’**operatore aritmetico**, infatti % si utilizza quando vogliamo trovare il resto di una divisione.

In questo caso occorre utilizzare l’operatore /.

Inoltre possiamo notare l’**errata indentazione** alla *riga 61*. Come nella funzione moltiplica, anche in questo caso non è stato gestito l’input da parte dell’utente. Una ulteriore modifica sarebbe da fare nel **data-type della divisione**, in quanto potrebbero esserci risultati che presentano la virgola, quindi occorre usare una variabile float (approssimata a 2 valori %.2f).

[FUNZIONE INS_STRING]

Questa funzione permette il semplice inserimento di una stringa di massimo 10 caratteri.

In questo caso non è stato implementato un **controllo sul numero di caratteri** inseriti da parte dell’utente. Questo è un errore che potrebbe causare un “*buffer overflow*” dove il dato inserito andrebbe a straripare nelle celle di memoria successive.

Una possibile soluzione, sarebbe il controllo dei caratteri inseriti da parte dell’utente oppure l’utilizzo della funzione fgets().