



# ASSISTENTE VIRTUALE

Esercizio 2

Mattia Pastorelli


```

1 #include <stdio.h>
2
3 void menu ();
4 void moltiplica ();
5 void dividi ();
6 void ins_string();
7
8
9 int main ()
10
11 {
12     char scelta = {'\0'};
13     menu ();
14     scanf ("%d", &scelta);
15
16     switch (scelta)
17     {
18         case 'A':
19             moltiplica();
20             break;
21         case 'B':
22             dividi();
23             break;
24         case 'C':
25             ins_string();
26             break;
27     }
28
29     return 0;
30 }
31
32
33 void menu ()
34 {
35     printf ("Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti\n");
36     printf ("Come posso aiutarti?\n");
37     printf ("A >> Moltiplicare due numeri\nB >> Dividere due numeri\nC >> Inserire una stringa\n");
38 }
39
40
41
42 void moltiplica ()
43 {
44     short int a,b = 0;
45     printf ("Inserisci i due numeri da moltiplicare:");
46     scanf ("%f", &a);
47     scanf ("%d", &b);
48
49     short int prodotto = a * b;
50
51     printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto);
52 }
53
54
55 void dividi ()
56 {
57     int a,b = 0;
58     printf ("Inserisci il numeratore:");
59     scanf ("%d", &a);
60     printf ("Inserisci il denominatore:");
61     scanf ("%d", &b);
62
63     int divisione = a % b;
64
65     printf ("La divisione tra %d e %d e': %d", a,b,divisione);
66 }
67
68
69
70
71 void ins_string ()
72 {
73     char stringa[10];
74     printf ("Inserisci la stringa:");
75     scanf ("%s", &stringa);
76 }
77
78

```

# Consegna:

- ◆ Per agire come un Hacker bisogna capire come pensare fuori dagli schemi. L'esercizio di oggi ha lo scopo di allenare l'osservazione critica. Dato il codice in allegato, si richiede allo studente di:
- ◆ -Capire cosa fa il programma senza eseguirlo
- ◆ -Individuare dal codice sorgente le casistiche non standard che il programma non gestisce (esempio, comportamenti potenziali che non sono stati contemplati)
- ◆ -Individuare eventuali errori di sintassi / logici
- ◆ -Proporre una soluzione per ognuno di essi




# Cosa fa il programma? (senza eseguirlo)

Il programma in questione è un  
Assistente Digitale.

E' costituito da un menù principale,  
nel quale vengono riportate tre  
funzioni dell'assistente:  
Moltiplicazione, Divisione,  
Inserimento di stringhe

L'utente può selezionare una di  
queste funzioni ed effettuare i calcoli  
richiesti.



-Individuare dal codice sorgente le casistiche non standard che il programma non gestisce (esempio, comportamenti potenziali che non sono stati contemplati)

Nel programma non sono stati contemplati:

- ◊ Un ciclo while nella prima parte del codice.
- ◊ Un ciclo If, Elif, Else
- ◊ La possibilità da parte dell'utente di poter ritornare al menu principale al termine dell'operazione o in caso di errore di digitazione
- ◊ Implementare l'ultima funzione stringa.



# Ciclo while nella prima parte del codice.

```
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     return 0;
29 }
```

Inserendo un ciclo «while» al posto dello «switch», permettiamo all'assistente digitale di non andare in crash in caso di scelta errata dell'utente.

Questo comando continuerà a proporre all'utente il Menu Iniziale, in caso venga digitato un carattere diverso da A o B o C. Il loop continua fino a quando l'utente non soddisferà la condizione imposta dal loop.

# Condizione If, Elif, Else

- ◈ Inserendo questa tipologia di condizione all'interno del ciclo while, l'assistente digitale darà all'utente ciò che viene richiesto, interrompendo il loop di richiesta del ciclo «While»

```
43 void multiplica ()
44 {
45     short int a,b = 0;
46     printf ("Inserisci i due numeri da moltiplicare:");
47     scanf ("%f", &a);
48     scanf ("%d", &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;
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);
78 }
```

```

1 #include <stdio.h>
2
3 void menu ();
4 void moltiplica ();
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             moltiplica();
19             break;
20         case 'B':
21             dividi();
22             break;
23         case 'C':
24             ins_string();
25             break;
26     }
27
28     return 0;
29 }
30
31
32
33 void menu ()
34 {
35     printf ("Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti\n");
36     printf ("Come posso aiutarti?\n");
37     printf ("A >> Moltiplicare due numeri\nB >> Dividere due numeri\nC >> Inserire una stringa\n");
38 }
39
40
41
42 void moltiplica ()
43 {
44     short int a,b = 0;
45     printf ("Inserisci i due numeri da moltiplicare:");
46     scanf ("%f", &a);
47     scanf ("%d", &b);
48
49     short int prodotto = a * b;
50
51     printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto);
52 }
53
54
55 void dividi ()
56 {
57     int a,b = 0;
58     printf ("Inserisci il numeratore:");
59     scanf ("%d", &a);
60     printf ("Inserisci il denominatore:");
61     scanf ("%d", &b);
62
63     int divisione = a % b;
64
65     printf ("La divisione tra %d e %d e': %d", a,b,divisione);
66 }
67
68
69
70 void ins_string ()
71 {
72     char stringa[10];
73     printf ("Inserisci la stringa:");
74     scanf ("%s", &stringa);
75 }
76
77
78

```

Possibilità da parte dell'utente di poter ritornare al menu principale al termine dell'operazione o in caso di errore di digitazione, scrivendo il comando adatto in C



# Implementare l'ultima funzione stringa.

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

Come da codice, si può notare che `ins_string` è una variabile fine a se stessa. Se eseguiamo il programma e premessimo C nel menù iniziale, ci chiederebbe di immettere una stringa.

Fino a qui tutto normale, ma guardando attentamente nel codice noteremmo che viene impostato un limite di caratteri di «10», quindi se si sforasse questo numero si potrebbe incorrere in errori di Buffer Over Flow.

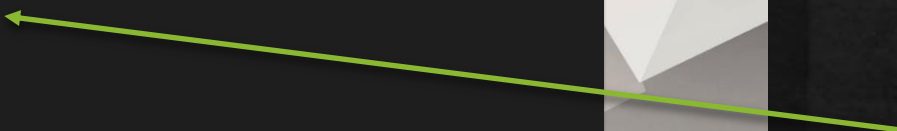
Nel caso venga rispettato il limite di caratteri, verrà mostrato a schermo la stringa ma senza uno scopo particolare.

Personalmente andrei ad implementare il numero di caratteri, impostando un limite più alto, con impossibilità di andare oltre nel caso l'utente inserisca più parole/numeri/simboli consentiti. Inoltre, farei in modo che l'assistente digitali ci possa aiutare con la stringa immessa, dandoci dei risultati di ricerca quali : Informazioni, numeri, pensieri ecc.



- Individuare eventuali errori di sintassi / logici
- Proporre una soluzione per ognuno di essi

```
1  #include <stdio.h>
2
3  void menu ();
4  void moltiplica ();
5  void dividi ();
6  void ins_string();
7
8
9  int main ()
10
11 {
12     char scelta = {'\0'};
13     menu ();
14     // Da %d a %c perchè Char
15     scanf ("%d", &scelta);
16
17     switch (scelta)
18     {
19         case 'A':
20             moltiplica();
21             break;
22         case 'B':
23             dividi();
24             break;
25         case 'C':
26             ins_string();
27             break;
28     }
29
30     return 0;
31 }
32
```



Possiamo notare che la variabile «Scelta» è preceduta da una tipologia «Char» (Caratteri), quindi nel comando scanf bisognerà adottare una funzione Char piuttosto che una funzione D (INT = interi).

Bisogna effettuare questo cambiamento per evitare problemi di conflitto nel programma e quindi che possa andare in Crash il sistema.

- Individuare eventuali errori di sintassi / logici
- Proporre una soluzione per ognuno di essi

Nella variabile «moltiplica» possiamo notare errori di sintassi, esaminandoli per ordine sono:

- La presenza di una tipologia «short int» per numeratore e denominatore, ma che non viene riportato correttamente negli Scanf.
- Dato che in una moltiplicazione l'utente potrebbe andare ad inserire numeri decimali, dobbiamo rendere possibile tale operazione, quindi andremo a sostituire «short int» con «float». Quest'ultimo termine ci permette di poter inserire numeri decimali, cosa che «short int» non ci permette.
- Cambieremo anche le funzioni, da «%d» a «%f» per non incorrere in eventuali errori
- Infine sul printf andremo a scrivere «%.2f» al posto dei «%d», questo permette all'utente di avere un risultato con massimo due numeri dopo la virgola.

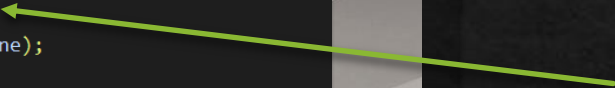
```
void moltiplica ()
{
    //Da short int a FLOAT
    short int a,b = 0;
    printf ("Inserisci i due numeri da moltiplicare:");
    scanf ("%f", &a);
    // Passare da "%d" a "%f"
    scanf ("%d", &b);

    // Da short int a FLOAT
    short int prodotto = a * b;

    // Da "%d" a "%.2f"
    printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto);
}
```

- Individuare eventuali errori di sintassi / logici
- Proporre una soluzione per ognuno di essi

```
0
1 void dividi ()
2 {
3     int a,b = 0;
4     printf ("Inserisci il numeratore:");
5     scanf ("%d", &a);
6     printf ("Inserisci il denominator:");
7     scanf ("%d", &b);
8
9     // % ci darà il resto dell'operazione, non il risultato.. quindi mettiamo "/"
10    int divisione = a % b;
11
12    printf ("La divisione tra %d e %d e': %d", a,b,divisione);
13 }
14
```



Nella variabile «dividi», possiamo individuare l'unico errore nella sezione «int divisione = a % b», in quanto il «%» restituisce solo il resto della divisione e non il risultato, quindi lo sostituiamo con l'operatore «/».

Si potrebbe pensare di implementare il codice con i Float e inserire una variabile «resto», per far sì che l'utente abbia il risultato e il resto dell'operazione.