

BUG HUNTING:

***STATO INIZIALE:***

Ci è stato richiesto di scovare eventuali bug e di apportare modifiche ad un programma scritto in C.

Il programma si presenta con un menù a scelta multipla, di cui 3 opzioni disponibili:

1) Moltiplicazione

2) Divisione

3) Inserimento di una stringa

Il programma non presenta la possibilità di riavviare il menù per effettuare altre operazioni.

```
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.
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);
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. }

```

### **PROBLEMI RISCONTRATI:**

- I. **PROBLEMI DI SCELTA:** a primo impatto lo switch case risulta incompleto, difatti utilizza (per quelle 3 opzioni che ci sono) una politica case sensitive, difatti se si prova ad inserire una delle opzioni con il carattere minuscolo, il programma terminerà automaticamente. Oltre a ciò anche l'inserimento di una delle lettere non presenti nello switch case porterà all'arresto del programma
- II. **ASSEGNAZIONI DI VARIABILI:** Un'altro errore riscontrato è l'errata assegnazioni di variabili su alcune funzioni, nella riga 45 si assegnano due variabili di tipo <<short int>> ma subito dopo si richiede di prendere, con uno scanf, un valore di tipo float, oltre alla stampa successiva in decimale <<%d>> anzichè <<%hd>>.

```

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

```

III.

Errore simile lo ritroviamo anche nella funzione << dividi >> e nella funzione << ins\_string >>.

Nella funzione `dividi` ad essere sbagliato è l'operatore utilizzato, difatti viene utilizzato l'operatore `<< % >>` che equivale al modulo ovvero al **resto** della divisione.

```
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. }
```

Mentre nella funzione `<< ins_string >>` viene inserito nel comando `scanf` un indirizzo di memoria `<< & >>` trattandosi di un'array questa operazione non è necessaria.

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

### ***FUNZIONALITA' MANCANTI:***

Nelle mancanze del programma possiamo sottolineare alcuni aspetti che rendono poco funzionale e pratico il codice, sia dal lato User experience che dal lato programmazione.

Vediamo innanzitutto la mancanza di parametri nei prototipi di funzione e di conseguenza la stessa mancanza nelle invocazioni di funzione in ambiente locale, questa mancanza per alcuni aspetti e in determinate circostanze non altera la normale esecuzione del codice, ma inserendole possiamo avere accesso a maggiori benefici, come gestione semplificata del codice ,difatti passando parametri formali e parametri attuali, possiamo modificare intere righe di codice

solo andando a modificare le variabili assegnate in ambiente locale o ancor meglio un futuro salvataggio dei risultati in ambiente locale, per poter riutilizzare quei dati in altre parti del codice.

Dal lato utente come citato all'inizio del report abbiamo principalmente 2 mancanze fondamentali :

- 1) La gestione iterativa dell'intero programma
- 2) Il controllo su inserimenti non corretti nel menù

### ***MIGLIORIA DEL CODICE : AGGIUNTE***

Il codice è stato modificato portando diverse implementazioni quali:

- I. ***Iterazioni per la gestione utente:*** Sono stati implementati 2 do-while concatenati che serviranno a gestire l'imput utente nel menù e la possibilità di riavviare il programma senza dover terminare l'esecuzione
- II. ***VARIABILI SENTINELLA:*** Sono state aggiunte 2 variabili sentinella per la gestione del do-while rendendo impossibile il problema dell'infinite loop.
- III. ***Case aggiuntivi :*** Sono stati implementati i case in Lower-cap per permettere all'utente di selezionare l'opzione richiesta sia con il block maiusc attivato che disattivato.
- IV. ***Parametri formali per la gestione delle funzioni:*** Sono stati aggiunti parametri formali (nel prototipo di funzione) , in modo tale che la funzione al momento dell'invocazione possa prendere in esame i parametri salvati in ambiente locale.
- V. ***Dichiarazione di variabili corrette:*** Nel corpo di funzione sono state modificate le variabili che entravano in conflitto, ed è stato impostato un nuovo print per la funzione dividi, infatti ora darà sia il risultato della divisione che il resto.
- VI. ***Controllo str:*** E' stato aggiunto un controllo sulla lunghezza della stringa, infatti se la stringa inserita supera i 10 caratteri il programma darà errore.

Di seguito riportato il codice modificato e gli screen di test:

***Codice:***

(PER UNA VISUALIZZAZIONE PIU' CHIARA DEL CODICE :

<https://pastebin.com/ygip7cj1>)

```
1. #include <stdlib.h> //AGGIUNTA DI STDLIB PER LA FUNZIONE EXIT
2. #include <stdio.h>
3. #include <string.h> //AGGIUNTA DI STRING.H PER IL CONTROLLO DELLA STRINGA
4. void menu ();
5. void moltiplica (float, float, float); //AGGIUNTA DEI PARAMETRI NOMINALI
6. void dividi (int,int,float,int); //AGGIUNTA DEI PARAMETRI NOMINALI
7. void ins_string(char[]); //AGGIUNTA PARAMETRO Nominale
8.
9.
10. int main ()
11.
12. {
13. int valore_sentinella; //AGGIUNTA DI UN VALORE SENTINELLA PER IL CONTROLLO DELL'INPUT E PER LA
CORRETTA ITERAZIONE.
14. char scelta = {'\0'};
15. //VARIABILI LOCALI PER LA FUNZIONE MOLTIPLICA
16. float MOLa,MOLb,RISmol;
17. //VARIABILI LOCALI PER LA FUNZIONE DIVIDI
18. int DIVa,DIVb,Resto_div;
19. float RISdiv;
20. //VARIABILI LOCALI PER LA FUNZIONE ins_string
21. char str[10] = {'\0'};
22.
23. //VARIABILI PER ITERAZIONE FINALE DEL PROGRAMMA
24. int scelta_finale, valore_sentinella2;
25.
26. do{ // MODELLO DI ITERAZIONE PER RIAVVIO APPLICAZIONE
27. menu ();
28.
29.
30.
31. do{ //MODELLO DI ITERAZIONE PER INPUT ERRATO
32. valore_sentinella = 0;
33. scelta_finale = 0;
34. valore_sentinella2 = 0;
35.
36. scanf ("%c", &scelta); //Non può essere preso in INPUT un valore decimale su una variabile di tipo
char, modifica con %c
37.
38. switch (scelta)
39. {
40. case 'A':
41. case 'a': // <---- Manca la scelta della lettera minuscola, aggiungerla per evitare problemi
42. moltiplica(MOLa,MOLb,RISmol);
43. break;
44. case 'B':
45. case 'b':
46. dividi(DIVa,DIVb,RISdiv,Resto_div); //Inserimento dei parametri LOCALI inerenti alla funzione.
47. break;
48. case 'C':
49. case 'c':
50. ins_string(str);
51. break;
52. //STRUTTURA DI CONTROLLO PER INPUT ERRATO CON VALORE SENTINELLA.
53. default:
54. valore_sentinella = 1;
55. break;
56. //FINE STRUTTURA DI CONTROLLO PER INPUT ERRATO.
```

```
57. }
58. }while(valore_sentinella == 1); //CHIUSURA DEL MODELLO DI ITERAZIONE
59. printf("\nHAI BISOGNO DI FARE ALTRE OPERAZIONI? .. \nPREMI 1 PER RIAVVIARE IL PROGRAMMA, ALTRO PER USCIRE\n");
60. scanf("\n%d", &scelta_finale);
61. if(scelta_finale == 1){
62.     valore_sentinella2 = 1;
63. }
64. else{
65.     exit;
66. }
67.
68. }while(valore_sentinella2 == 1);
69. return 0;
70.
71. }
72.
73.
74. void menu ()
75. {
76.     printf ("Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti\n");
77.     printf ("Come posso aiutarti?\n");
78.     printf ("A >> Moltiplicare due numeri\nB >> Dividere due numeri\nC >> Inserire una stringa\n");
79.
80. }
81.
82.
83. void moltiplica (float a, float b, float prodotto) // AGGIUNTA DELLE VARIABILI E CAMBIO DI TIPO ALLA FUNZIONE
84. {
85.     a,b = 0;
86.     printf ("Inserisci i due numeri da moltiplicare:\n");
87.     scanf ("%f", &a);
88.     scanf ("%f", &b);
89.
90.     prodotto = a * b;
91.
92.     printf ("Il prodotto tra %.2f e %.2f e': %.2f", a,b,prodotto);
93. }
94.
95.
96. void dividi (int a, int b, float divisione, int modulo) // AGGIUNTA DELLE VARIABILI E CAMBIO DI TIPO ALLA FUNZIONE
97. {
98.     a,b = 0;
99.     printf ("Inserisci il numeratore:");
100.    scanf ("%d", &a);
101.    printf ("Inserisci il denominatore:");
102.    scanf ("%d", &b);
103.    divisione = a / b;
104.    modulo = a % b;
105.    printf("\nLa divisione tra %d e %d e' : %.2f", a,b, divisione);
106.    printf ("\nIl resto della divisione tra %d e %d e': %d", a,b,modulo); //CAMBIO DELLA FUNZIONE DA DIVISIONE A RESTO (SCOPO DEL MODULO)
107. }
108.
109.
110.
```



05/11/22, 14:56

Pastebin.com - Printed Paste ID: <https://pastebin.com/>

```
111.
112.
113. void ins_string (char stringa[]) // AGGIUNTA DELLE VARIABILI E CAMBIO DI TIPO ALLA FUNZIONE
114. {
115.     do{
116.         printf ("Inserisci la stringa\n:");
117.         scanf ("%s", stringa);
118.         if(strlen(stringa)> 10){
119.             printf("\nINSERIRE UNA STRINGA DI MASSIMO 10 CARATTERI");
120.         }
121.     }while(strlen(stringa) > 10);
122.     printf("\nLA STRINGA INSERITA E' : %s", stringa);
123. }
```

## TEST DEL PROGRAMMA:

```
(kali㉿kali)-[~/Desktop]
$ ./ProgrammaOttimizzato
Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti
Come posso aiutarti?
A >> Moltiplicare due numeri
B >> Dividere due numeri
C >> Inserire una stringa
a
Inserisci i due numeri da moltiplicare:
12
45
Il prodotto tra 12.00 e 45.00 e': 540.00
HAI BISOGNO DI FARE ALTRE OPERAZIONI? ..
PREMI 1 PER RIAVVIARE IL PROGRAMMA, ALTRO PER USCIRE
1
Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti
Come posso aiutarti?
A >> Moltiplicare due numeri
B >> Dividere due numeri
C >> Inserire una stringa
b
Inserisci il numeratore:12
Inserisci il denominatore:45

La divisione tra 12 e 45 e' : 0.00
Il resto della divisione tra 12 e 45 e': 12
HAI BISOGNO DI FARE ALTRE OPERAZIONI? ..
PREMI 1 PER RIAVVIARE IL PROGRAMMA, ALTRO PER USCIRE
1
Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti
Come posso aiutarti?
A >> Moltiplicare due numeri
B >> Dividere due numeri
C >> Inserire una stringa
c
Inserisci la stringa
:Andrea

LA STRINGA INSERITA E' : Andrea
HAI BISOGNO DI FARE ALTRE OPERAZIONI? ..
PREMI 1 PER RIAVVIARE IL PROGRAMMA, ALTRO PER USCIRE
f
```

In questa prima fase di test andiamo a vedere il comportamento del programma con i caratteri lower-case , come possiamo notare vengono accettati, inoltre possiamo anche l'ottimizzazione del programma con la nuova implementazione

alla funzione << dividi >> e la richiesta all'utente di effettuare altre operazioni.

<< **NB** : Per tutte le operazioni aritmetiche è buona norma utilizzare il float o il double, dato che la probabilità di lavorare con un numero reale è alta, in questo esercizio si sono presi in considerazione diversi tipi di dato solo per allenare e dimostrare le conoscenze ottenute a lezione.>>

```
(kali@kali)~[~/Desktop]
$ ./ProgrammaOttimizzato
Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti
Come posso aiutarti?
A >> Moltiplicare due numeri
B >> Dividere due numeri
C >> Inserire una stringa
F
Q
U
12
C
Inserisci la stringa
:ANDREAANDREAANDREAANDREA

INSERIRE UNA STRINGA DI MASSIMO 10 CARATTERIInserisci la stringa
:█
```

In questa parte di test andiamo a notare i "vincoli" messi all'utente, ovvero se inseriamo un carattere nella scelta del menù diverso dalle opzioni date, il programma continuerà a richiedere l'input corretto all'utente, stessa situazione per il controllo della stringa, se l'utente inserirà una stringa maggiore di 10 caratteri, apparirà un errore che richiederà nuovamente l'input della stringa.