

Progetto S2L5

PROGRAMMAZIONE IN C – 🐞 BUG HUNTING

Dato il codice:

- 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

```

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
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

```

```

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 }
79

```

Capire cosa fa il programma senza eseguirlo

Leggendo il codice possiamo capire che questo svolge delle operazioni algebriche. È suddiviso in quattro funzioni *void*: una di menù, e le restanti 3 per le operazioni da svolgere. In ogni funzione sono presenti dei comandi input per la scelta dell'operazione da effettuare, e per i numeri da inserire.

Il codice è inteso per svolgere un'operazione e poi terminare con il comando *break*.

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

- Il codice presenta, nella funzione *void menu*, solamente 3 opzioni relative alla scelta da effettuare (lettere A,B,C). Non sono prese in considerazioni altre risposte da classificare come errate o non valide (es. lettere diverse, lettere uguali ma minuscole, numeri): in caso di un input diverso da quelli proposti, il programma cesserà.
- Analogamente, nelle funzioni di operazione algebrica, non sono presenti istruzioni per eventuali input diversi da numeri, che se inseriti, causeranno la cessazione del programma.
- Anche la funzione *void ins_strin* non presenta istruzioni qualora l'input fosse una stringa di oltre 10 caratteri o diversa da un valore di testo.

Individuare eventuali errori di sintassi / logici

```
int main ()  
  
{  
    char scelta = {'\0'};  
    menu ();  
    scanf ("%d", &scelta);
```

- La variabile scelta è di tipo *char*, rappresentante un singolo carattere. Tuttavia nell'input il formato selezionato è "%d", utilizzato per rappresentare un numero intero. Il valore da utilizzare in questo caso è "%c", come illustrato in figura a sinistra.

```
void moltiplica ()  
{  
    short int a,b = 0;  
    printf ("Inserisci i due numeri da moltiplicare:");  
    scanf ("%f", &a);  
    scanf ("%d", &b);  
  
    short int prodotto = a * b;  
  
    printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto);  
}
```

- Analogo problema si presenta nella funzione *moltiplica*, dove il formato dell'input del primo numero da moltiplicare dovrebbe essere "%d" invece di "%f".

```
void moltiplica ()  
{  
    short int a,b = 0;  
    printf ("Inserisci i due numeri da moltiplicare:");  
    scanf ("%d", &a);  
    scanf ("%d", &b);  
  
    short int prodotto = a * b;  
  
    printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto);  
}
```

- Nella variabile troviamo *short int* troviamo il formato "%d". Questo è da correggere in "%hd" in quanto formato corrispondente al valore *short int*.

```
int main ()  
  
{  
    char scelta = {'\0'};  
    menu ();  
    scanf ("%d", &scelta);
```

```
void moltiplica ()  
{  
    short int a,b = 0;  
    printf ("Inserisci i due numeri da moltiplicare:");  
    scanf ("%hd", &a);  
    scanf ("%hd", &b);  
  
    short int prodotto = a * b;  
  
    printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto);  
}
```

```
void moltiplica ()  
{  
    short int a,b = 0;  
    printf ("Inserisci i due numeri da moltiplicare:");  
    scanf ("%hd", &a);  
    scanf ("%hd", &b);  
  
    short int prodotto = a * b;  
  
    printf ("Il prodotto tra %hd e %hd e': %hd", a,b,prodotto);  
}
```

Individuare eventuali errori di sintassi / logici

```
void dividi ()  
{  
    int a,b = 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);  
}
```

- Nella funzione *dividi* viene erroneamente utilizzato l'operatore modulo "%". L'operatore corretto da utilizzare è invece "/". Correggiamo inoltre l'indentazione.

```
void dividi ()  
{  
    int a,b = 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);  
}
```

- La variabile inserita è un numero intero *int*. È preferibile inserire un numero decimale in modo da essere più precisi nel calcolo. Correggiamo modificando la variabile in *float* inserendo *%.2f* per il risultato. In questo caso abbiamo un numero decimale a 2 cifre dopo la virgola.

```
void dividi ()  
{  
    int a,b = 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 dividi ()  
{  
    int a,b = 0;  
    printf ("Inserisci il numeratore:");  
    scanf ("%d", &a);  
    printf ("Inserisci il denominatore:");  
    scanf ("%d", &b);  
  
    float divisione = (float) a / b;  
  
    printf ("La divisione tra %d e %d e': %.2f", a,b,divisione);  
}
```

```
void ins_string ()  
{  
    char stringa[10];  
    printf ("Inserisci la stringa:");  
    scanf ("%s", &stringa);  
}
```

- Nella variabile *ins_string* abbiamo un errore di sintassi nell'input. La & non è necessaria. La variabile non presenta inoltre alcuna azione nel caso il valore inserito ecceda i 10 simboli o caratteri o numeri inseriti. Lo correggiamo con messaggio di errore nel caso di input superiore ai 10 caratteri.

```
void ins_string ()  
{  
    char stringa[11];  
    printf ("Inserisci la stringa (max 10 caratteri:");  
    scanf ("%s", stringa);  
    if (strlen(stringa) >10)  
    {  
        printf("Input non valido.\n");  
        void ins_string ();  
    }  
    else  
    {  
        printf("Hai inserito: %s\n", stringa);  
    }  
}
```

```
File Actions Edit View Help
└──(giacomo㉿kali)-[~/Desktop]
$ ./Consegna Consegnando
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:
3
6
Il prodotto tra 3 e 6 e': 18
└──(giacomo㉿kali)-[~/Desktop]
$ ./Consegna
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:9
Inserisci il denominatore:2
La divisione tra 9 e 2 e': 4.50
```

```
└──(giacomo㉿kali)-[~/Desktop]
$ ./Consegna
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 (max 10 caratteri):CarloMagno
Hai inserito: CarloMagno
└──(giacomo㉿kali)-[~/Desktop]
$ ./Consegna
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 (max 10 caratteri):AlessandroMagno
Input non valido.
zsh: segmentation fault  ./Consegna
```

Eseguiamo il file: il programma è ora in grado di eseguire le operazioni richieste. Il risultato della divisione è con 2 numeri decimali come richiesto. Abbiamo anche il messaggio di errore in seguito a input maggiore di 10 caratteri nella terza funzione.