

BUG HUNTING

Nell'esercizio viene proposto un codice sorgente all'interno del quale si dovranno individuare gli errori sintattici e/o logici presenti e correggerli.

Struttura del codice

Il programma eseguito dal codice vuole emulare un'assistente digitale che, su input dell'utente, esegue alcune operazioni aritmetiche e non.

Le azioni proposte all'utente in fase di avvio del programma sono:

- A) Moltiplicazione
- B) Divisione
- C) Inserimento di una stringa di testo

Viene richiesto di digitare **A**, **B** o **C** per selezionare l'opzione desiderata e procedere quindi con l'operazione o l'inserimento della stringa.

All'interno del codice vengono richiamate una funzione **main** e 4 funzioni **void**. La peculiarità di quest'ultima funzione è che, a differenza della prima, non restituisce alcun valore, motivo per cui è semanticamente corretto concluderla con **return;**. La funzione main termina invece con **return 0;**

Principali bug e casistiche non affrontate dal codice

1. Nella funzione main, si evidenzia un'incongruenza tra la variabile dichiarata, "scelta" (di tipo **char**), e l'argomento della funzione scanf corrispondente (che non dovrebbe essere %d ma %c)

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

2. Le opzioni di scelta iniziali sono A, B e C ma non viene considerata l'eventualità che l'utente possa digitare questi 3 caratteri in versione **minuscola**. Infatti, provando ad eseguire il programma digitando le lettere a, b o c il programma non le prende in input e termina:

```
(kali㉿kali)-[~/Desktop]
$ ./Esercizio_con_bug
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
(kali㉿kali)-[~/Desktop]
$
```

Pertanto, ho inserito alcune righe di codice affinché fossero contemplate le casistiche in cui l'utente scelga le opzioni proposte in formato lowercase.

```
switch (scelta)
{
    case 'A':
        moltiplica();
    case 'a':
        moltiplica();
        break;
    case 'B':
        dividi();
    case 'b':
        dividi();
        break;
    case 'C':
        ins_string();
    case 'c':
        ins_string();
        break;
}
```

Successivamente ho verificato l'effettivo funzionamento del programma, ricompilando e rieseguendo il file:

```
(kali㉿kali)-[~/Desktop]
$ ./Esercizio_corretto
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: 
```

Inoltre, ho introdotto un controllo sull'inserimento degli input per far visualizzare all'utente un messaggio di "arrivederci" prima che il programma termini:

```
int main ()
{
    // >> Moltiplicare due numeri
    // >> Dividere due numeri
    // >> Inserire una stringa
    char scelta= {'\0'};
    menu ();
    scanf ("%c", &scelta);

    if (scelta != 'A' && scelta != 'a' && scelta != 'B' && scelta != 'b' && scelta != 'C' && scelta != 'c') {
        printf ("A presto!");
        exit;;
    }

    switch (scelta) {
        case 'A':
            {Esercizio_corretto.c -- Esercizio_corretto
            moltiplica();
            case 'a':
            moltiplica();
            **MENU PRINCIPALE**
            break;
            case 'B':
            Benvenuto! Sono un assistente digitale. Posso aiutarti a sbrigare alcuni compiti!
            Come posso aiutarti?
            dividi();
            case 'b':
            dividi();
            Scegli una tra le opzioni proposte; digita qualsiasi altro tasto per uscire
            break;
            case 'C':
            ins_string();
            case 'c':
            ins_string();
            break;
        }
    }
    Inserisci la stringa: tiofreddo
    return 0;
}
```

```
(kali@kali)-[~/Desktop]
$ ./Esercizio_corretto
A presto!
***MENU PRINCIPALE***

Benvenuto! Sono un assistente digitale. Posso aiutarti a sbrigare alcuni compiti!
Come posso aiutarti?

Scegli una tra le opzioni proposte; digita qualsiasi altro tasto per uscire

A >> Moltiplicare due numeri
B >> Dividere due numeri
C >> Inserire una stringa
4
A presto!
```

3. All'interno della funzione **void moltiplica ()** a e b sono variabili dichiarate di tipo "short int", ma è consigliabile utilizzare il tipo int in quanto ha a disposizione più bit di memoria. Inoltre se a è una variabile di tipo int, l'argomento della funzione scanf corrispondente dovrebbe essere %d e non %f

```

void moltiplica ()
{
    short int a,b = 0; // Per la casistica in questione è preferibile utilizzare il tipo int
    printf ("Inserisci i due numeri da moltiplicare:");
    scanf ("%f", &a); // Errore: la variabile "a" è di tipo int, mentre il tipo di argomento specificato dalla funzione scanf è di tipo float
    scanf ("%d", &b);

    short int prodotto = a * b; // Per la casistica in questione è consigliabile il tipo int
    printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto);
}

```

Pertanto, ho apportato le seguenti modifiche

```

void moltiplica ()
{
    int a,b = 0;
    printf ("\nInserisci i due numeri da moltiplicare:");
    printf ("\n\nPrimo fattore: ");
    scanf ("%d", &a);
    printf ("\nSecondo fattore: ");
    scanf ("%d", &b);
    int prodotto = a * b;
    printf ("\nIl prodotto tra %d e %d e': %d", a,b,prodotto);
}

```

4. L'operazione di divisione presenta l'operatore aritmetico sbagliato (% indica il modulo); l'operatore corretto è /. Inoltre non viene data la possibilità all'utente di ottenere il risultato corretto di divisioni che possono produrre un numero **decimale**

```
int divisione = a % b;
```

```
printf ("La divisione tra %d e %d e': %d", a,b,divisione);
```

Pertanto, ho modificato il tipo di variabile precedentemente assegnato alla divisione da int a **float** eseguendo anche un casting sulle variabili corrispondenti ai due dividendi; ho inoltre modificato l'argomento specificato dalla funzione "divisione" da %d a **&f** per consentire l'eventuale stampa corretta di un numero decimale come risultato della divisione:

```

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

    float divisione = (float)a / (float)b;

    printf ("La divisione tra %d e %d e': %f", a,b,divisione);

    return;
}

```

Successivamente ho testato il codice inserito, provando ad eseguire un'operazione il cui risultato atteso è un numero decimale:

```

(kali@kali)-[~/Desktop]
$ ./Esercizio_corretto
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: 5
Inserisci il denominatore: 2
La divisione tra 5 e 2 e': 2.500000

```

5. Infine, nella funzione **void ins_string()** relativa all'opzione di scelta di inserimento di un testo, ho inserito la possibilità di stampare la stringa inserita

```

void ins_string ()
{
    char stringa[10];
    printf ("\nInserisci la stringa: ");
    scanf ("%s", &stringa);
    printf ("Stringa inserita: %s ", stringa);
}

```

```
(kali㉿kali)-[~/Desktop]
```

```
$ ./Esercizio_corretto
```

```
***MENU PRINCIPALE***
```

Benvenuto! Sono un assistente digitale. Posso aiutarti a sbrigare alcuni compiti!
Come posso aiutarti?

Scegli una tra le opzioni proposte; digita qualsiasi altro tasto per uscire

```
printf("\nInserisci la stringa: ");
```

A >> Moltiplicare due numeri);

B >> Dividere due numeri inserita: %s ", stringa);

C >> Inserire una stringa

c

Inserisci la stringa (Lunghezza max 10 caratteri): Ciao

Stringa inserita: Ciao

```
(kali㉿kali)-[~/Desktop]
```

```
$
```

Read File

Where Is

Replace

Cut

Paste

Execute

Justify

Wrote