

S2/L5 - Osservazione critica

Gianmarco Mazzoni / NetRaiders, 08 marzo 2024

Consegna:

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.

Il codice allegato è il seguente, sono evidenziati in rosso i punti in cui ho notato **porzioni di sintassi errata**, mentre in blu sono indicati miei commenti che suggeriscono **aggiunte per gestire comportamenti non contemplati**:

```
#include <stdio.h>

void menu ();
void moltiplica ();
void dividi ();
void ins_string();

int main ()

{
    char scelta = {'\0'}; //La dichiarazione è superflua
    menu ();
    scanf ("%d", &scelta);

    switch (scelta)
    {
        // Nessuna gestione per inserimenti in minuscolo
        case 'A':
            moltiplica();
            break;
        case 'B':
            dividi();
            break;
        case 'C':
            ins_string();
            break;
    } //Nessuna gestione in caso di errore di input
```

```
return 0;
```

```
}
```

```
void menu ()
```

```
{
```

```
    printf ("Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare  
alcuni compiti\n"); //Errore di battitura.
```

```
    printf ("Come posso aiutarti?\n");
```

```
    printf ("A >> Moltiplicare due numeri\nB >> Dividere due numeri\nC >>  
Inserire una stringa\n");
```

```
}
```

```
void moltiplica ()
```

```
{
```

```
    short int  a,b = 0;    //Un valore è null, mentre l'altro viene  
dichiarato.
```

```
                        //Inoltre, usare uno short è insensato.
```

```
    printf ("Inserisci i due numeri da moltiplicare:");
```

```
    scanf ("%f", &a);      //%f non può funzionare, non essendo float.
```

```
                        //Cambiare in %d
```

```
    scanf ("%d", &b);
```

```
    short int prodotto = a * b;
```

```
                        //Nuovamente, usare short per un prodotto è  
insensato.
```

```
    printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto); //\n mancante.
```

```
}
```

```
void dividi ()
```

```
{
```

```
    int  a,b = 0;          //Un valore è null, mentre l'altro viene  
dichiarato.
```

```
    printf ("Inserisci il numeratore:");
```

```
    scanf ("%d", &a);
```

```
    printf ("Inserisci il denominatoro:");
```

```
    scanf ("%d", &b);
```

```

int divisione = a % b;
//Questo operatore non divide, ma restituisce il resto.
//Assente gestione della divisione per zero.
//Inoltre, usare int per una divisione può causare errori di calcolo.

printf ("La divisione tra %d e %d e': %d", a,b,divisione); //\n
mancante.
//
}

void ins_string ()
{
    char stringa[10]; //Dimensione troppo corta, può causare errori.
    printf ("Inserisci la stringa:");
    scanf ("%s", &stringa); //& ridondante
    //Output della stringa inserita mancante. Insensato.
}

```

Per motivi di sicurezza e a scopo di analisi, il codice non è stato compilato e/o eseguito. Io e il mio team abbiamo revisionato il codice della consegna, che presenta gravi mancanze, in particolare:

- **Nessuna gestione in caso di scelta errata dell'Utente;**
- **Nessuna gestione per calcoli impossibili;**
- **Diversi errori di sintassi che compromettono il funzionamento.**

Abbiamo trovato delle soluzioni ai problemi rilevati durante l'analisi. Seguono dettagli.

1. Gestione della scelta

Durante la selezione, all'Utente è richiesto di inserire una delle lettere tra A, B e C. Sfortunatamente, il programma non sarebbe né in grado di rilevare se la lettera inserita è minuscola, e ciò causerebbe un errore, né di gestire l'errore umano, ovvero l'inserimento di una lettera diversa da quelle indicate.

Non volendo aggiungere ulteriori librerie per gestire questa casistica, abbiamo semplicemente aggiunto un case che includesse una lettera minuscola, e, in caso di input errato, stampa un avviso che chiede un riavvio all'Utente.

In verde sono evidenziate le modifiche necessarie.

```

switch (scelta)
{

```

```

case 'A':
case 'a':
    moltiplica();
    break;
case 'B':
case 'b':
    dividi();
    break;
case 'C':
case 'c':
    ins_string();
    break;
default:
    printf("Scelta sbagliata, chiusura in corso...\n
           Si prega di riavviare scegliendo tra le lettere
           disponibili!\n");
}

```

2. Dividere per zero

Nella funzione di divisione del programma, vengono presi in input due numeri interi, con lo scopo di dare in output il risultato della divisione del primo per il secondo.

Innanzitutto, nel programma viene utilizzato l'operando % che non esegue un'operazione di divisione, ma restituisce il resto. Nonostante ciò, l'operazione di divisione sarebbe risultata errata, nel caso di risultato non intero, per l'assenza dei (float) nel codice.

Ultimo punto, ma non per importanza, nell'input del secondo dato sarebbe stato uguale a zero, sarebbe stato impossibile dividere, e avrebbe causato problemi al programma.

Abbiamo quindi risolto aggiungendo un if che dopo l'input dell'Utente verifica se il numero inserito equivale a 0, in quel caso, non verrà effettuato alcun calcolo e darà in output che non è possibile dividere per zero, terminando poi il programma. Altrimenti, l'output conterrà il risultato, più 2 numeri decimali, dopo aver eseguito la conversione delle variabili in float.

In verde sono evidenziate le modifiche necessarie.

```

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

    if(b==0){
        printf("Non si può dividere per 0!\n");
    }
}

```

```

}
else{
    float divisione = (float)a/(float)b;
    printf ("La divisione tra %d e %d e' : %.2f\n", a, b, divisione);
}
}

```

3. Casi di sintassi errata

Abbiamo notato un numero non indifferente di syntax errors, e casi in cui il codice si sarebbe potuto scrivere meglio. Seguono gli esempi trovati, già indicati in rosso nella prima parte.

```

char scelta = {'\0'};          //La dichiarazione del carattere non è necessaria

scanf ("%d", &scelta);        //Non avrebbe funzionato, va cambiato con %c

printf ("Inserisci i due numeri da moltiplicare:");
scanf ("%f", &a);              //%f non può funzionare, non essendo variabile
float

short int a,b = 0;             //Un valore è null, mentre l'altro viene
dichiarato.
//Inoltre usando una variabile short si rischia di non contenere il valore in
input

short int prodotto = a * b;    //Stessa cosa per il prodotto

int a,b = 0;                   //Come prima, un valore è null, mentre l'altro viene
dichiarato.

int divisione = a % b;         //Questo operatore non divide, serve /
//Inoltre, usare int per una divisione può causare errori di calcolo.

char stringa[10];              //Dimensione troppo corta, può causare errori.
//Inoltre manca l'Output della stringa inserita mancante l'input è insensato.

```

Conclusione e correzione del codice.

```

#include <stdio.h>

void menu ();
void moltiplica ();

```

```

void dividi ();
void ins_string();

int main ()

{
    char scelta = '\0';
    menu ();
    scanf (" %c", &scelta);

    switch (scelta)
    {
        case 'A':
        case 'a':
            moltiplica();
            break;
        case 'B':
        case 'b':
            dividi();
            break;
        case 'C':
        case 'c':
            ins_string();
            break;
        default:
            printf("Scelta sbagliata, chiusura in corso...\nSi prega di
riavviare scegliendo tra le lettere disponibili!\n");
    }

    return 0;

}

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

```

```

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

    int prodotto = a * b;

    printf ("Il prodotto tra %d e %d e': %d\n", a,b,prodotto);
}

```

```

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

    if(b==0){
        printf("Non si può dividere per 0!\n");
    }

    else{
        float divisione = (float)a/(float)b;
        printf ("La divisione tra %d e %d e' : %.2f\n", a, b, divisione);

    }

}

```

```

void ins_string ()
{
    char stringa[100];
    printf ("Inserisci la stringa:");
    scanf (" %99[^\n]s", stringa); //Per leggere tutta la stringa se include
spazi
    printf("Hai inserito: %s\n", stringa);
}

```

