

Bug Hunting e Ottimizzazione del programma

Il programma

Prendendo in analisi il codice fornito, il programma rappresenta un assistente digitale che propone tre opzioni differenti all'utente aiutandolo con dei compiti.

La prima è di moltiplicare due numeri facendo inserire all'utente due numeri per restituirne il prodotto, la seconda di dividere due numeri restituendo il resto della divisione e per terzo di inserire una stringa.

Visualizzazione del codice errato

```
File Actions Edit View Help
#include <stdio.h>

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

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

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

    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 ()
{
    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);
}

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

Identificazione degli errori nel codice sorgente

-In questo caso, nella funzione *main()* e nella costruzione di *switch* il programma gestisce soltanto tre opzioni di input che sono A,B o C ; di conseguenza non sono state gestite tutte le casistiche in cui l'utente può inserire una scelta differente.

Inoltre poiché *scelta* è una variabile di tipo *'char'* che memorizza un singolo carattere si dovrebbe utilizzare *%c*. Quindi, per leggere correttamente un singolo carattere da input e assegnarlo alla variabile *scelta*, dobbiamo utilizzare *%c* nella chiamata a *scanf()*.

```
File Actions Edit View Help
void menu ();
void multiplica ();
void dividi ();
void ins_string();

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

    switch (scelta)
    {
        case 'A':
            multiplica();
            break;
        case 'B':
            dividi();
            break;
        case 'C':
            ins_string();
            break;
    }

    return 0;
}
```

Nella funzione *multiplica()* *'a'* e *'b'* vengono dichiarate come variabili di tipo *<short int>* ma nella funzione *scanf* è stato utilizzato *<%f>* per ricevere l'input. Poiché i numeri vengono letti come *float* e *int*, la funzione non gestisce correttamente il caso in cui l'utente inserisce numeri decimali anziché interi.

Da notare che *<short int>* è un numero da 2byte e vi è da considerare la possibilità che il prodotto possa essere maggiore.

```
void multiplica ()
{
    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);
}
```

In questo caso nella funzione *dividi()* non è gestito il caso in cui il divisore sia 0, il che potrebbe causare un errore durante l'esecuzione del programma.

Un evidente errore di sintassi è che è stato utilizzato l'operatore '%' anziché l'operatore di divisione e quindi produrrà inevitabilmente un risultato errato.

Inoltre le variabili *<int>* possono gestire solo numeri interi senza parte decimale, il che potrebbe portare a una perdita di precisione nei calcoli di divisione quando si trattano numeri con 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);
}
```

Nella funzione *ins_string* non viene specificata una lunghezza massima per la stringa che può essere letta dall'utente. Di conseguenza se l'utente inserisce una stringa più lunga di quanto sia allocato lo spazio di memoria per "*stringa*", si verifica un buffer overflow che può rappresentare una vulnerabilità pericolosa in quanto un eventuale attaccante può approfittare di questo errore prendendo il controllo del programma; di conseguenza è necessaria una lunghezza massima per la stringa che scanf può leggere.

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

Risoluzione dei problemi nel codice :

Funzione *main()*:

Per correggere l'errore nella lettura dell'input dell'utente, sostituire la chiamata a `scanf("%d", &scelta)`; con `scanf(" %c", &scelta)`;.. Questo assicura che venga letto un singolo carattere da input e assegnato correttamente alla variabile scelta.

Inoltre si può realizzare un ciclo `while` che ripeta la richiesta finché l'input non è tra le opzioni valide ('A', 'B' o 'C') per evitare le casistiche non gestite.

Funzione *moltiplica()*:

Modificare la dichiarazione delle variabili `a` e `b` in `int` anziché `float` e `int`, rispettivamente.

Inoltre, utilizzare `%d` anziché `%f` nella chiamata a `scanf()` per leggere entrambi i numeri come interi.

Aggiungere un controllo per gestire il caso in cui l'utente inserisca numeri decimali anziché interi. Se l'utente inserisce un numero decimale, la moltiplicazione potrebbe produrre risultati non desiderati o imprevisti, poiché il programma aspetta input interi.

Funzione *dividi()*:

Sostituire l'operazione `a % b` con `a / b` per calcolare correttamente il risultato della divisione tra `a` e `b`.

Aggiungere una condizione per gestire il caso in cui il divisore (`b`) sia uguale a zero. Se `b` è zero, visualizzare un messaggio appropriato all'utente e interrompere l'esecuzione della funzione; è importante includere una condizione che controlli se il divisore è zero e gestire questo caso in modo appropriato.

Cambiare le variabili nella funzione `dividi()` da `int` a `double` per consentire di gestire numeri con la virgola (decimali) anziché solo numeri interi. Questo rende il programma più flessibile e versatile, consentendo all'utente di eseguire divisioni con risultati decimali anziché solo interi.

Funzione *ins_string()*:

Modificare la chiamata a `scanf()` per leggere la stringa utilizzando `%10s` anziché `%s`. Questo limita il numero massimo di caratteri letti a 10, evitando overflow del buffer.

Aggiungere un controllo per gestire il caso in cui l'utente inserisca una stringa più lunga di 10 caratteri.

Nel programma, ho incluso la possibilità per l'utente di uscire dal programma in qualsiasi momento attraverso l'aggiunta dell'opzione 'X' nel menu principale. Questa funzionalità è stata implementata all'interno del ciclo principale del programma utilizzando un'istruzione `do-while`. Il programma continua a mostrare il menu e ad eseguire le operazioni selezionate finché l'utente non seleziona l'opzione 'X' per uscire.

Visualizzazione del codice corretto

```
File Actions Edit View Help
#include <stdio.h>
void menu();
void multiplica();
void dividi();
void ins_string();

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

        switch (scelta) {
            case 'A':
                multiplica();
                break;
            case 'B':
                dividi();
                break;
            case 'C':
                ins_string();
                break;
            case 'X':
                printf("Uscita dal programma. Arrivederci!\n");
                return 0; // Uscita dal programma
            default:
                printf("Input non valido. Si prega di selezionare A, B, C o X.\n");
                break;
        }
    } while (scelta != 'X');

    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");
    printf("X >> Uscire dal programma\n");
}

void multiplica() {
    int a, b;
    printf("Inserisci i due numeri da moltiplicare: ");
    scanf("%d %d", &a, &b); // Legge due numeri interi da input

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

void dividi() {
    double a, b; // Dichiarate come double per consentire numeri con virgola
    printf("Inserisci il numeratore: ");
    scanf("%lf", &a); // Utilizza %lf per leggere un double
    printf("Inserisci il divisore: ");
    scanf("%lf", &b);

    if (b == 0.0) {
        printf("Errore: divisione per zero non consentita.\n");
        return;
    }

    double risultato = a / b;
    printf("Il risultato della divisione tra %.2lf e %.2lf è: %.2lf\n", a, b, risultato);
}

// Funzione per inserire una stringa di massimo 10 caratteri
void ins_string() {
    char stringa[11];
    printf("Inserisci una stringa di massimo 10 caratteri: ");
    scanf("%10s", stringa); // Legge una stringa di massimo 10 caratteri

    printf("Hai inserito la stringa: %s\n", stringa);
}
```

TEST sul terminale di KALI-LINUX

```
kali@kali: ~/Desktop
File Actions Edit View Help
(kali@kali)-[~/Desktop]
$ ./codicefinale
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
X >> Uscire dal programma
A
Inserisci i due numeri da moltiplicare: 5 5
Il prodotto tra 5 e 5 è: 25
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
X >> Uscire dal programma
B
Inserisci il numeratore: 9
Inserisci il divisore: 2
Il risultato della divisione tra 9.00 e 2.00 è: 4.50
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
X >> Uscire dal programma
C
Inserisci una stringa di massimo 10 caratteri: Matteo
Hai inserito la stringa: Matteo
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
X >> Uscire dal programma
X
Uscita dal programma. Arrivederci!
```