Bug Hunting

Studente: Emanuele Di Stefano

Struttura generale

-Inclusione delle Librerie Il programma inizia con l'inclusione della libreria standard stdio.h, necessaria per svolgere le operazioni di input e output, come la lettura da tastiera e la scrittura su schermo.

-Dichiarazione delle Funzioni Prima della funzione main(), ci sono dichiarazioni delle funzioni menu(), moltiplica(), dividi(), e ins_string(). Queste dichiarazioni sono prototipi di funzione e servono a indicare al compilatore i tipi di funzione che verranno utilizzate prima che esse vengano effettivamente definite.

-Funzione main() La funzione main() è il punto di ingresso del programma. Qui, la logica principale del programma viene eseguita, inclusa la gestione del menu di opzioni e il controllo del flusso a seconda della scelta dell'utente.

Funzionalità delle Funzioni

-Funzione menu() Questa funzione stampa un messaggio di benvenuto e le istruzioni per l'utente, presentando le opzioni disponibili: moltiplicare due numeri, dividere due numeri, e inserire una stringa. È una funzione puramente informativa e non ritorna alcun valore.

-Funzione moltiplica() Dopo aver ricevuto l'input dell'utente nella funzione main(), se l'utente sceglie di moltiplicare due numeri, questa funzione viene chiamata. Richiede all'utente di inserire due numeri e poi calcola il prodotto di questi, stampando il risultato. Gestisce anche la corretta inizializzazione e lettura dei dati.

-Funzione dividi() Analogamente alla funzione moltiplica(), questa funzione gestisce la divisione. Chiede due numeri all'utente (numeratore e denominatore) e calcola il quoziente, assumendo che il denominatore non sia zero. Gestisce anche gli errori di input, come la divisione per zero.

-Funzione ins_string() Questa funzione permette all'utente di inserire una stringa, che viene poi stampata a schermo. È importante per gestire le operazioni di input di testo da parte dell'utente

e dimostra come manipolare le stringhe in C.

Analisi e Correzione Script in C

Il compito richiesto mira a testare e affinare le nostre capacità di osservazione critica, cruciali in campo informatico e, soprattutto, nella sicurezza delle informazioni. Il codice fornito per questo esercizio gestisce un semplice menu interattivo che permette agli utenti di scegliere tra moltiplicare due numeri, dividere due numeri, o inserire una stringa. La nostra missione è di comprendere a pieno il codice, individuare gli errori e migliorarlo dove necessario.

Analisi degli Errori e Soluzioni

1. **Errore nella lettura della variabile scelta**

Descrizione dell'errore: La variabile scelta è dichiarata come char, che è corretto visto che rappresenta un singolo carattere ('A', 'B', 'C'). Tuttavia, quando viene letta dall'input dell'utente, si utilizza %d, un specificatore di formato per gli interi, anziché %c, che è destinato ai caratteri.

Soluzione: Modificare la chiamata a scanf per utilizzare %c e così leggere correttamente il carattere inserito dall'utente.

```
scanf(" %c", &scelta);
```

L'aggiunta di uno spazio prima del %c aiuta a ignorare eventuali spazi bianchi o newline lasciati nel buffer.

2. **Errore di inizializzazione in moltiplica()**

Descrizione dell'errore: Le variabili a e b dovrebbero essere inizializzate a zero per evitare comportamenti non definiti. Nell'attuale implementazione, solo b viene inizializzato.

Soluzione: Inizializzare entrambe le variabili a zero quando vengono dichiarate.

```
short int a = 0, b = 0;
```

3. **Formato scorretto in moltiplica()**

Descrizione dell'errore: Quando si leggono i numeri per la moltiplicazione, si utilizza %f per a, che è un formato destinato ai numeri in virgola mobile, mentre a è un intero.

Soluzione: Utilizzare %d per leggere un intero.

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

4. **Operatore errato in dividi()**

Descrizione dell'errore: Viene usato l'operatore % che calcola il resto di una divisione, anziché / per ottenere il quoziente della divisione.

Soluzione: Sostituire % con / nella formula di calcolo.

```
int divisione = a / b;
```

5. **Gestione scorretta delle stringhe in ins_string()**

Descrizione dell'errore: Utilizzare &stringa in scanf non è corretto perché stringa è già un indirizzo di un array.

Soluzione: Modificare la chiamata a scanf eliminando l'operatore &.

```
scanf("%s", stringa);
```

Correzione Script:

Correzione Script (commenti + simpatiche aggiunte)

```
#include <stdio.h>
void menu();
void moltiplica();
void dividi();
void ins_string();
int main()
{



    char scelta = ’\0’;
    menu();
    // Corretto: Uso di %c per leggere un carattere e inclusione di uno spazio prima per scartare
whitespace residui.
    scanf(" %c", &scelta);
    switch (scelta)
    {

        case ’A’:
        moltiplica();
        break;      case
        ’B’:
        dividi();
        break;
        case ’C’:
            ins_string();
            break;
        default:
            printf("Opzione non valida.\n");
            break;
    }
    return 0;

}
void menu()
{

    printf("*****Benvenuto*****, sono Emanuele, posso aiutarti a sbrigare (al posto tuo :D) alcuni
compiti.\n\n");
    printf("Come posso aiutarti? (posso dividere o moltiplicare, le somme o le sottrazioni falle da
solo)\n\n\n");
    printf("A >> Moltiplicare due numeri\n\nB >> Dividere due numeri\n\nC >> Inserire una stringa\n\n");
}
void moltiplica()
{

    int a = 0, b = 0; // Corretto: Modifica tipo variabile da short int a int e inizializzazione corretta.
    printf("Inserisci i due numeri da moltiplicare:"); // Corretto: Sostituzione di %f con %d per la
    corretta lettura di valori interi. scanf("%d", &a); scanf("%d", &b); int prodotto = a * b; //
    Corretto: Uso di int anzich



                                                        é short int per evitare overflow.
    printf("Il prodotto tra %d e %d è: %d\n", a, b, prodotto);
```

```
}
void dividi()
{

    int a = 0, b = 0;
    printf("Inserisci il numeratore:");
    scanf("%d", &a);
    printf("Inserisci il denominatore:");
    scanf("%d", &b);
    // Aggiunto controllo per prevenire la divisione per zero.
    if (b != 0) {

        int quoziente = a / b; // Corretto: Sostituito % con / per calcolare il quoziente.
        printf("La divisione tra %d e %d è: %d\n", a, b, quoziente);
    } else {
        printf("Errore: divisione per zero non permessa.\n");
    }
}
void ins_string()
{

    char stringa[10];
    printf("Inserisci la stringa:");
    // Corretto: Rimosso & dall'argomento di scanf perché 'stringa' è già un indirizzo.
    scanf("%9s", stringa); // Aggiunto limite di lettura per prevenire overflow del buffer.
    printf("Hai inserito: %s\n", stringa);
}
```