Progetto S2-L5

Analisi e risoluzione problemi del codice in C

Traccia

Per agire come un Hacker bisogna capire come pensare fuori dagli schemi. L'esercizio di oggi ha lo scopo di allenare l'osservazione critica. 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

Prima parte del codice

Una volta aperto il documento, ho fatto un'analisi sul codice leggendo e annotando tutti gli eventuali errori del programma.

/* In azzurro ho direttamente commentato il codice segnalando eventuali errori e proponendo soluzioni al fine di correggere e migliorare l'esecuzione del programma*/

Seconda parte del codice

Nel codice ho riscontrato sia errori logici, sia di sintassi e addirittura anche ortografici. Ho riportato anche in quest'immagine tutte le mie considerazioni direttamente con un commento nel file di testo.

A questo punto ho applicato direttamente sul codice le soluzioni a cui ho pensato

Prima parte del codice

Così come ho fatto per la fase di analisi, ho fatto per la fase di miglioramento, ovvero ho commentato tutte le modifiche direttamente sul file in azzurro. Qui si può notare tutto il corpo modificato della funzione main che adesso ha un ciclo e quindi l'utente può ripetere le operazioni fintanto che desidera farlo.

```
  GNU nano 7.2                                                progetto2.c *
#include <stdio.h>

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

int main() {
    char scelta = '\0';      // ho rimosso le parentesi graffe

/* ho inserito un ciclo do-while in modo di darà la possibilità all'utente di scegliere se procedere
   con un'altra operazione oppure terminare il programma al termine di ogni operazione. Ho inoltre
   implementato anche le minuscole, un caso default e il caso D,d per terminare il programma */
    do {
        menu();
        scanf(" %c", &scelta);          // ho cambiato con " %c"

        switch (scelta) {
            case 'A':
                moltiplica();
                break;
            case 'a':
                moltiplica();
                break;
            case 'B':
                dividi();
                break;
            case 'b':
                dividi();
                break;
            case 'C':
                ins_string();
                break;
            case 'c':
                ins_string();
                break;
            case 'D':
                printf("Arrivederci!\n");
                break;
            case 'd':
                printf("Arrivederci!\n");
                break;
            default:
                printf("Scelta non valida. Riprova.\n");
        }
    } while (scelta != 'D' && scelta != 'd');   // il ciclo si conclude se l'utente inserisce D,d

    return 0;
}

^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute       ^C Location      M-U Undo         M-A Set Mark     M-] To Bracket
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify       ^/ Go To Line    M-E Redo         M-6 Copy         ^Q Where Was
```

Seconda e terza parte del codice

Qui mi sono occupato di modificare eventuali errori ortografici, modifiche ai tipi delle variabile dedicate alle operazioni e miglioramento della funzione per la stringa che avrebbe creato grossi problemi di esecuzione se non fossi intervenuto. Inoltre ho aggiunto il caso di errore se si prova a dividere per 0 e un printf di conferma avvenuta immissione della stringa

```
 GNU nano 7.2                                                progetto2.c *
// correzione della parola "assistente"
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\nD >> Uscire\n");
}


// come indicato ho modificato i valori affinché si possano svolgere operazioni con numeri reali
void moltiplica() {
    float a, b;
    printf("Inserisci i due numeri da moltiplicare: ");
    scanf("%f%f", &a, &b);

    float prodotto = a * b;

    printf("Il prodotto tra %.2f e %.2f e': %.2f\n", a, b, prodotto);
}

/* stessa cosa per quanto riguarda la divisione e ovviamente modifica di % con /
   ho anche inserito un messaggio di errore se si prova a dividere per 0 */
void dividi() {
    float a, b;
    printf("Inserisci il numeratore: ");
    scanf("%f", &a);
    printf("Inserisci il denominatore: ");
    scanf("%f", &b);

    if (b != 0) {
        float divisione = a / b;
        printf("La divisione tra %.2f e %.2f e': %.2f\n", a, b, divisione);
    } else {
        printf("Errore: divisione per zero!\n");
    }
}

/* ho lasciato la dimensione massima della stringa a 10 ma ho fatto comparire un messaggio di avviso
   del limite massimo consentito di caratteri all'utente. In questo modo se si immettono più di 9
   caratteri il programma ignora i caratteri oltre il nono. Ho leggermente migliorato la funzione
   inserendo un piccolo ciclo che mi permette di pulire il buffer di input ed evitare che i caratteri
   in eccesso vadano in input non appena compare di nuovo il menù. Ho inserito un printf per far
   visualizzare a schermo la stringa inserita fungendo da conferma di corretta esecuzione*/
void ins_string() {
    char stringa[10];
    printf("Inserisci la stringa di massimo 9 caratteri: ");
    scanf("%9s", stringa);

    int c;
    while ((c = getchar()) != '\n' && c != EOF);

^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute       ^C Location      M-U Undo         M-A Set Mark     M-] To Bracket
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify       ^/ Go To Line    M-E Redo         M-6 Copy         ^Q Where Was
    int c;
    while ((c = getchar()) != '\n' && c != EOF);

    printf("Ecco la stringa che hai inserito: %s\n", stringa);
}

^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute       ^C Location      M-U Undo         M-A Set Mark     M-] To Bracket
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify       ^/ Go To Line    M-E Redo         M-6 Copy         ^Q Where Was
```

Per questo progetto è tutto, tanti saluti

Amedeo Natalizi