# Consegna S2/L5 CyberSecurity (Progetto Week-End 1)

Riccardo Agostino Monti

# Iniziamo il debugging.

Per iniziare il debugging la prima cosa che dobbiamo fare e capire che cosa il programma dovrebbe fare.
Leggendo il programma si capisce che si è voluto realizzare un assistente digitale capace di eseguire 3 operazioni per noi :

1. Fare la moltiplicazione tra 2 numeri.
2. Fare la divisione tra 2 numeri.
3. Inserire una stringa.

# Errori di sintassi.

Il prossimo passo è correggere gli errori di sintassi.
Gli errori di sintassi trovati sono soprattutto all'interno della scanf e printf visto che alcuni formati inseriti (mi riferisco a %d per esempio) sono errati.

Inoltre durante l'inizializzazione delle variabili, sia nella funzione moltiplica() che dividi() solo la B veniva inizializzata a 0, come sappiamo questo può portare errori non desiderati in fase di esecuzione del programma.

Sono anche presenti alcuni errori grammaticali

# Errori logici

Il prossimo passo è analizzare tutti gli errori logici del programma proposto.
L'unico errore logico trovato è presente nella funzione ins_string() .
In questo caso il programma farà inserire la stringa ma non dà nessun responso, quindi l'utente non riceve nessuna conferma per l'inserimento della stringa.

Inoltre dovendo gestire stringhe nel codice è opportuno pulire l'input ogni volta che esso viene eseguito, per farlo sfruttiamo la funzione rewind(stdin)

Altro errore logico è l'assenza di controlli nella divisione per 0.

# Esaminazione dei comportamenti non gestiti.

Il programma presenta diversi comportamenti non gestiti. La maggior parte di questi sono errori che potrebbero presentarsi se l'utente inserisce un carattere non presente tra le scelte. Lo stesso problema si presenta anche nel caso in cui l'utente inserisca una scelta valida ma con un carattere minuscolo non riconosciuto dallo switch.

E' inoltre consigliato realizzare un ciclo do-while per far chiudere il programma sotto scelta dell'utente (aggiungendo la scelta 'D' allo switch) per permettere di fare più azioni prima di uscire.

Anche la gestione dell'input delle stringhe lo farei in modo diverso utilizzando la funzione fgets e impostando un limite alla quantità di caratteri che possono essere inseriti da tastiera.

Per gestire tutto questo è necessario inserire le librerie stdbool.h e string.h.

Infine si consiglia di aggiungere la libreria stdlib.h per pulire il terminale utilizzando la funzione system("clear") ogni volta che si avvia il programma e si effettua una scelta e utilizzare la funzione <<rewind (stdin) >> utilizzata per pulire gli input. Infine alla fine del completamento di ogni funzione aggiungere la funzione <<getchar()>> per mandare in pausa il terminale e chiedere un input da tastiera per continuare. Per ultima aggiungiamo la libreria

# Esaminazione dei comportamenti non gestiti.

Inoltre è stata aggiunta una nuova funzione clearBuffer() che nel caso in cui i caratteri inseriti nell'inserimento della stringa sono più di quelli massimi consentiti pulisce il buffer dai caratteri rimanenti. E salva solo i primi 9 (Il decimo carattere salvato è sempre \0)

A seguire gli screen del codice commentato con gli errori e consigli per migliorare il codice.

```c
#include <stdio.h> //Dovremmo includere anche le libreria stdbool.h , string.h, system.h per migliorare il codice
void menu ();
void moltiplica ();
void dividi ();
void ins_string();


int main () //Nel main aggiungerei un ciclo do-while per far chiudere il programma sotto richiesta e per gestirne gli errori.

{
        char scelta = {'\0'};
        menu ();
        scanf ("%d", &scelta); //Scelta è di tipo char e il formato per la scanf + %c non %d

        switch (scelta) //Nello switch aggiungerei un case 'D' per chiedere al utente se vuole uscire dal programma senza compiere operazioni.
        {                       //Inoltre aggiungerei anche un case per le lettere minuscole così da gestire anche i casi in cui un utente dà quelle in input.
                                //Inoltre manca anche un case default per gestire gli errori di input dell'utente.
                case 'A':
                moltiplica();
                break;
                case 'B':
                dividi();
                break;
                case 'C':
                ins_string();
                break;
        }

return 0;

}
```

A seguire gli screen del codice commentato con gli errori e consigli per migliorare il codice.

```c
void menu ()
{
        printf ("Benvenuto, sono un assitente 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"); //Aggiungere D >> Esci dal programma

}


//All'inizio di ogni funzione, prima di eseguire qualsiasi comando userei system("clear") per pulire il terminale
//Alla fine di ogni funzione userei getchar() per aspettare un input da tastiera prima di continuare.


void moltiplica ()
{
        short int  a,b = 0; //Cambiare il tipo di dato in float per dare la possibilità di far inserire inoltre in questo modo si inizializza solo' b' e non ' a'
        printf ("Inserisci i due numeri da moltiplicare:");
        scanf ("%f", &a);   //Formato sbagliato, se si vuole utilizzare il formato per gli short int è %hd
        scanf ("%d", &b);   //Mentre se vogliamo cambiarlo in float basta cambiare il formato della scanf di a da %d a %f

        short int prodotto = a * b; //Anche qui cambiare il tipo in float.

        printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto); //Cambiare il formato dei dato in %f
}
```

A seguire gli screen del codice commentato con gli errori e consigli per migliorare il codice.

```c
void dividi ()
{
        int  a,b = 0; //Anche qua imposterei float il tipo di dato per dare la possibilità di inserire numeri decimali
        printf ("Inserisci il numeratore:");
        scanf ("%d", &a); //Se si cambia il tipo di dato bisogna cambiare anche il formato in %f
        printf ("Inserisci il denumeratore:");
        scanf ("%d", &b); //Se si cambia il tipo di dato bisogna cambiare anche il formato in %f

        int divisione = a % b; //Cambiare il tipo della divisione in float. Inoltre l'operatore utilizzato è % e non /

        printf ("La divisione tra %d e %d e': %d", a,b,divisione); //Cambiare i formati
}



void ins_string ()
{
        //Gestire meglio l'imput della stringa utilizzando la funzione fgets della libreria string.h e stampare la stringa inserita
        //per confermare l'inserimento.
        char stringa[10];
        printf ("Inserisci la stringa:");
        scanf ("%s", &stringa);
}
```

Si nota che nella funzione dividi() manca il controllo per la divisione per 0

Infine questo è la modifica suggerita del codice:

```c
#include <stdio.h>
#include<stdbool.h>
#include <string.h>
#include<stdlib.h>

void menu ();
void moltiplica ();
void dividi ();
void ins_string();
void clearBuffer();
int main ()

{
        bool  uscita;
        char scelta = {'\0'};

        do{
                menu ();
                scanf ("%c", &scelta);

                switch (scelta)
                {
                        case 'A':
                         moltiplica();
                         break;

                        case 'B':
                         dividi();
                         break;

                        case 'C':
                         ins_string();
                         break;

                        case 'D':
                        uscita=true;
                        break;

                        case 'a':
                         moltiplica();
                         break;

                        case 'b':
                         dividi();
```

Infine questo è la modifica suggerita del codice:

```
                        case 'b':
                         dividi();
                         break;

                        case 'c':
                         ins_string();
                         break;

                        case 'd':
                        uscita=true;
                        break;

                        default:
                        printf("\n\nErrore, scelta non valida, riprova\n");
                        break;
                }
                scelta = '\0';
        }while(scelta!='a' && scelta!='A' && scelta!='b' && scelta!='B' && scelta!='c' && scelta!='C' && scelta!='d' && scelta!='D' && !uscita);

return 0;

}


void menu ()
{
        system("clear");
        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 >> Esci dal programma\n-->  ");

}
```

# Infine questo è la modifica suggerita del codice:

```c
void moltiplica ()
{
        system("clear");
        float  a = 0,b = 0;
        printf ("Inserisci il primo fattore da moltiplicare: ");
        scanf ("%f", &a);
        rewind(stdin);
        printf ("Inserisci il secondo fattore da moltiplicare: ");
        scanf ("%f", &b);
        rewind(stdin);

        float prodotto = a * b;

        printf ("Il prodotto tra %.3f e %.3f e': %.4f\n\n", a,b,prodotto);
        printf ("Premere un tasto per continuare...");
        getchar();
        getchar();
}


void dividi ()
{
        system("clear");
        float  a = 0, b = 0;
        printf ("Inserisci il numeratore:");
        scanf ("%f", &a);
        rewind(stdin);
        do{
                printf ("Inserisci il denominatore:");
                scanf ("%f", &b);
                rewind(stdin);
                if(b==0)
                printf("\nNon puoi dividere per 0, inserisci un altro numero!!\n");
        }while(b==0);
        float divisione = a / b;

        printf ("La divisione tra %f e %f e': %f\n\n", a,b,divisione);
        printf ("Premere un tasto per continuare...");
        getchar();
        getchar();

}
```

Infine questo è la modifica suggerita del codice:

```c
void ins_string ()
{
        system("clear");
        char stringa[10];
        printf ("Inserisci la stringa:");
        getchar();
        fgets (stringa, 10, stdin);
        printf ("\n\nQuesta è la tua stringa: %s\n\n", stringa);
        printf ("Premere un tasto per continuare...");
        clearBuffer();
        getchar();

}

void clearBuffer()
{
    int c;
    while ((c = getchar()) != '\n' && c != EOF)
    {
        // Scarta i caratteri residui nel buffer di input
    }
}
```

# Conclusioni

Ho modificato il codice come deciso.
Il programma così risulta corretto, completamente funzionante e con una gran quantità di casistiche coperte.
Ho risolto alcuni bug di input (a volte le scanf leggevano il comando di invio sul terminale e quindi il programma andava avanti in errore) utilizzando la funzione getchar() ho risolto questo problemino, un altro modo per risolverlo era aggiungere una spazio prima del formato di dato nella scanf ma risultava più “sporco”.