Bug Hunting

Obiettivo: Osservare il codice di un programma, appuntare eventuali errori e risolverli

1) Presentazione del codice ed Osservazione

Per cominciare ho scaricato il file Txt dell'esercizio della settimana 2. Il primo obiettivo che mi è stato richiesto di fare è quello di leggerlo e capire di cosa si tratta senza però eseguirlo.

```
#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 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");

}
```

Escludendo le variabili che vengono dichiarate con il tipo ***void***, posso notare che il codice è innanzitutto diviso in due parti. Nell'immagine in alto si presenta l'Istruzione Condizionale Switch che viene usata per dare istruzioni con la Variabile ***char scelta*** e che a seconda dell'input digitato, in questo caso A, B o C, indirizza al ***case*** con l'operazione scelta. I ***break*** alla fine di ogni ***case*** sono utili per uscire automaticamente da un'operazione una volta eseguita e non andare ad eseguire quella successiva. In basso c'è un ***void menu*** che è usato probabilmente come sottoprocesso per mostrare una sorta di Interfaccia Grafica, o meglio che mi mostra un messaggio di benvenuto insieme alle possibili scelte che suggeriscono la digitazione dei caratteri che avviano lo switch.
La seconda parte del codice mostra invece le operazioni a cui conducono i ***case*** dello switch a seconda di quale carattere viene digitato, ovvero **Moltiplica** e **Dividi** che sono delle semplici operazioni matematiche e **Inserisci una Stringa**, in questo caso senza un particolare scopo ma se il codice venisse aggiornato e quindi ampliato potrebbe essere utilizzata ad esempio come un **"Inserimento Nome"**.

```
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 denumeratore:");
        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);
}
```

2) Individuare casistiche non standard

Ho ricercato possibili problemi che il codice potrebbe avere inizialmente non a livello di sintassi ma di istruzioni.

- Il primo che mi è saltato all'occhio è che nello ***switch*** non è presente l'istruzione ***default***, ovvero nel caso non si digiti un carattere che sia A, B o C il controllo del programma viene indirizzato all'istruzione successiva che in questo caso è la chiusura dello stesso programma;
- Un altro appunto da fare all'interno di ***switch*** sarebbe preferibile inserire anche le versioni minuscole delle tre lettere usate nei ***case*** in maniera tale che non è necessario digitare per forza la lettera in maiuscolo;
- Un altro caso può essere similmente al precedente se nelle operazioni **Moltiplica** e **Dividi** utilizziamo un carattere alfabetico invece di un numero, ci sarebbe la chiusura del programma;
- Infine può essere più che un problema una scomodità, il fatto che una volta terminata un'operazione il programma non chieda se si vuole avviarne una nuova, perciò bisogna avviare ripetutamente il programma ogni volta;
- Ultimo caso può essere di immettere nella stringa un numero di caratteri maggiore di quelli assegnati, ovvero 10, ciò porterebbe a un errore.

3) **Eventuali errori di sintassi e logici**

Per il terzo obiettivo ho fatto una copia del file Txt dell'esercizio per mantenere l'originale. In questo punto mi è stato chiesto di trovare eventuali errori di sintassi e logici. Gli errori di sintassi che ho trovato sono:

- In **int main** ho modificato **%d** in scanf in **%c** perchè **char scelta** si riferisce a un carattere e non ad un numero;

```
int main ()

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

- In **Moltiplica** ho modificato **%f** (che si riferisce a **float)** della variabile "a" in **%d.** Ho inoltre modificato **short int** in **int** perchè avrei dovuto sostituire ogni **%d** in **%hd** e visto che sono simili nel numero di byte che hanno a disposizione ho cancellato una sola parola per praticità.

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

        prodotto = a * b;

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

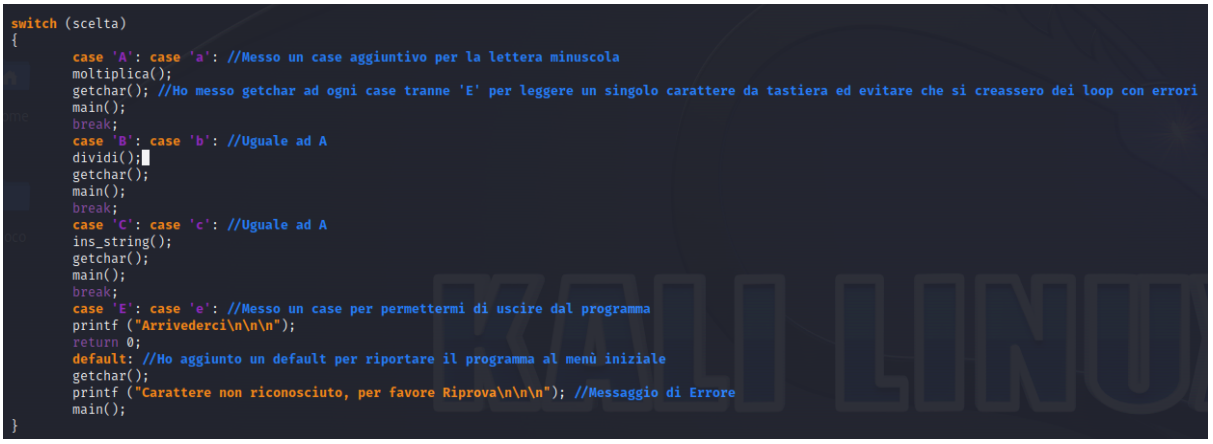
- In **Dividi** ho cambiato l'operatore **" % "** che restituisce solo il resto di una divisione con l'operatore **" / "** che invece ci fornisce il risultato della divisione stessa.

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

        divisione = a / b;

        printf ("La divisione tra %d e %d e': %f", a, b, divisione);
}
```

Note: ho dichiarato le variabili di **Moltiplica** e **Dividi** all'inizio delle operazioni ma anche lasciandole come nell'originale non cambia nulla.
Successivamente ho provato a compilare ed avviare il codice dell'esercizio appena corretto su ***Kali Linux*** senza trovare nessun tipo di Errore di Sintassi o di Esecuzione. Fino a questo punto non ho ancora modificato il codice secondo le considerazioni del fatte nel punto **2**.

4) Proporre una soluzione

Ho cercato di riscrivere il codice proposto con qualche variante per rimediare a tali considerazioni. Ho allegato il File.c insieme al report ma vado ad illustrare le modifiche fatte:



```
switch (scelta)
{
        case 'A': case 'a': //Messo un case aggiuntivo per la lettera minuscola
        moltiplica();
        getchar(); //Ho messo getchar ad ogni case tranne 'E' per leggere un singolo carattere da tastiera ed evitare che si creassero dei loop con errori
        main();
        break;
        case 'B': case 'b': //Uguale ad A
        dividi();
        getchar();
        main();
        break;
        case 'C': case 'c': //Uguale ad A
        ins_string();
        getchar();
        main();
        break;
        case 'E': case 'e': //Messo un case per permettermi di uscire dal programma
        printf ("Arrivederci\n\n\n");
        return 0;
        default: //Ho aggiunto un default per riportare il programma al menù iniziale
        getchar();
        printf ("Carattere non riconosciuto, per favore Riprova\n\n\n"); //Messaggio di Errore
        main();
}
```

- Innanzitutto nello ***switch*** ho aggiunto un ***case*** ad ognuna delle principali scelte in modo che si potesse eseguire un'operazione anche digitando caratteri in minuscolo;
- Ho aggiunto un case per scegliere di uscire direttamente dal programma digitando **"E"** o **"e"**;
- Sempre nello switch ho aggiunto un ***default*** in maniera tale che digitando qualsiasi altro carattere o numero ci riportasse all'inizio del programma e in aggiunta un messaggio di Errore;
- Ho impostato su tutti i ***case*** tranne "E" un ***getchar()*** per leggere un solo carattere da tastiera e "azzerare" gli input (senza di esso ogni volta che chiedevo di fare una nuova operazione mi dava errore e dava un valore automatico alle varianti dopo);
- Infine ho messo in ogni ***case*** un **main()** in maniera tale che alla fine di ogni operazione il programma tornasse al menù iniziale;

```
void ins_string ()
{
        char stringa[10];
        printf ("Inserisci la stringa:");
        scanf ("%s", &stringa);
        printf ("Ciao %s\n\n\n", &stringa); //Messaggio aggiuntivo alla compilazione della stringa
}
```

- Ho aggiunto anche un piccolo messaggio una volta immessa la stringa.

Purtroppo non sono riuscito a trovare una soluzione al problema del digitare caratteri nelle operazioni di Moltiplica e Dividi oltre al problema della stringa troppo lunga (a quest'ultimo problema potremmo ovviare in maniera spartana aumentando il numero di caratteri inseribili).