MARIO MARSICANO

LABORATORIO DI EPICODE

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:

1. Capire cosa fa il programma senza eseguirlo.
2. Individuare dal codice sorgente le casistiche non standard che il programma non gestisce (esempio, comportamenti potenziali che non sono stati contemplati).
3. Individuare eventuali errori di sintassi / logici.
4. Proporre una soluzione per ognuno di essi.

1. Possiamo intuire già dal primo momento cosa va a fare il programma leggendo la funzione **void menu ()** descrivendo già le azioni che andrà eventualmente ad eseguire.

Troviamo un messaggio di benvenuto con le scelte che invita ad effettuare in base alla problematica da risolvere creando 3 casistiche diverse: un operazione di moltiplicazione, un operazione di divisione ed una di inserimento stringa.

In base alla selezione proposta, raggiungibile da una dei tre caratteri (A per eseguire una moltiplicazione di due numeri – B per eseguire la divisione tra due numeri e C per scrivere una stringa non più grande di 10 caratteri) il programma effettuerà l'operazione.

2. Nel programma non è stata contemplata una procedura di errore in caso di scelte sbagliate dal menu principale, inoltre non c'è una procedura di eventuale proposta di eseguire altre operazioni ed una richiesta di chiusura.
3. Di seguito elenchiamo gli errori riscontrati, presentatoci al momento della consegna del codice:

1° Errore

```
12        char scelta = {'\0'};
13        menu ();
14        scanf ("%d", &scelta);
15
16        switch (scelta)
17        {
18                case 'A':
19                moltiplica();
20                break;
21                case 'B':
22                dividi();
23                break;
24                case 'C':
25                ins_string();
26                break;
27        }
```

Possiamo notare il primo errore di sintassi: avendo dichiarato una funzione di tipo CHAR alla riga 12 e quindi di tipo alfabetico , va ad indirizzare il tipo dell'argomento di tipo numerico **%d** come vediamo in foto quando poi invece avrebbe dovuto indirizzare l'argomento che richiamava la funzione di tipo carattere e cioè **%c**

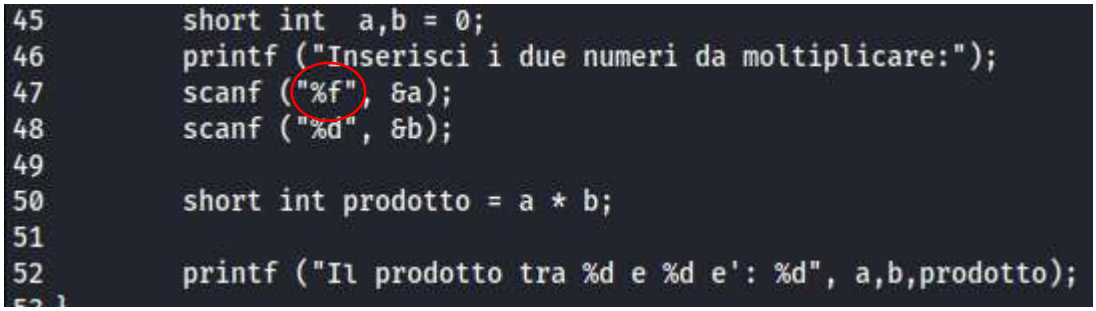
2° Errore

```
12        char scelta = {'\0'};
13        menu ();
14        scanf ("%d", &scelta);
15
16        switch (scelta)
17        {
18                case 'A':
19                moltiplica();
20                break;
21                case 'B':
22                dividi();
23                break;
24                case 'C':
25                ins_string();
26                break;
27        }
```

Il secondo errore lo possiamo riscontrare nella logica del programma: nelle tre casistiche di scelta iniziale non è possibile avere un ulteriore opzione che indichi che bisogna scegliere solo tra le opzioni proposte e che quindi non è possibile scegliere un opzione al di fuori delle tre menzionate nel menu principale.

Questo lo possiamo fare con la funzione di **default** che, appunto, da un messaggio d'errore in caso di una qualsiasi scelta errata all'infuori delle prime tre e che quindi non combacino con la scelta A oppure B oppure C.

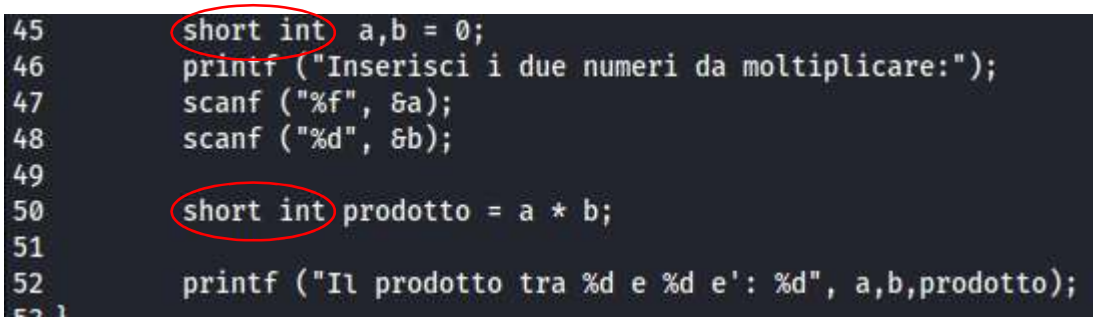
3° Errore



```
45        short int  a,b = 0;
46        printf ("Inserisci i due numeri da moltiplicare:");
47        scanf ("%f", &a);
48        scanf ("%d", &b);
49
50        short int prodotto = a * b;
51
52        printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto);
```

Possiamo notare un altro errore di sintassi del linguaggio nella riga 47 nella quale dichiara un argomento di tipo **FLOAT %f** (che indica le variabili numeriche reali) invece di dichiararla di tipo %d che richiama i numeri interi per svolgere l'operazione; infatti sulla riga 48 notiamo la notazione giusta che indica l'argomento di tipo **%d**.

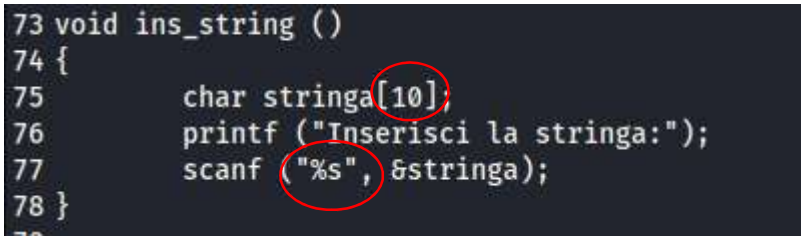
4° Errore



```
45        short int  a,b = 0;
46        printf ("Inserisci i due numeri da moltiplicare:");
47        scanf ("%f", &a);
48        scanf ("%d", &b);
49
50        short int prodotto = a * b;
51
52        printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto);
```

Possiamo notare un altro errore di sintassi del linguaggio nelle righe 45 e 50 dove vengono dichiarate delle variabili per il calcolo di tipo **Short int** che rappresenta un tipo di numero intero che occupa meno spazio in memoria e che quindi può rappresentare meno numeri invece di mettere una variabile di tipo **int** in modo che si possa fare il calcolo della moltiplicazione con più numeri disponibili e con un range molto più ampio

5° Errore



L'ultimo errore riscontrato si trova nella riga 77 della funzione Void in_string(): è stato creato un Array di tipo CHAR contrassegnando il suo valore tra le parentesi quadre che vediamo nella riga 75, in quanto è stato messo un argomento di tipo %s per inserire la stringa, che è corretto, ma in questo modo l'Array non è stato limitato perché proprio nella riga 75 viene dichiarata una variabile di tipo CHAR che crea un Array di 8 bit e nelle parentesi quadre sono stati assegnati 10 bit; questo comporta un errore di Buffer Stack Overflow in quanto il programma va ad allocare 2 bit in più in quella variabile, e quindi essendoci altri 6 bit disponibili, questi sarebbero inutilizzati in quella memoria ed allocabili con dei puntatori esterni che potrebbero richiamare codici o parte di codici malevoli. Avrebbe dovuto mettere un limitatore di stringa **{\0}** nella riga 77 in modo da assegnare solo i 10 bit dichiarati nella variabile mettendo il giusto limite in quell'area

4. Alla luce degli errori riscontrati, si propone di seguito il codice riscritto e corretto:

CODICE CORRETTO

```
// Assistente digitale



#include <stdio.h>

//Variabili Void
void menu ();
void moltiplica ();
void dividi ();
void ins_string();

//Casistiche per la scelta
int main ()

{
        char scelta = {'\0'};
```

```
	menu ();
	scanf ("%c", &scelta);
	switch (scelta)
	{
		case 'A':
		moltiplica();
		break;
		case 'B':
	 dividi();
	 break;
		case 'C':
	 ins_string();
	 break;
	 default:
	 printf ("Opzione non riconosciuta!");
	 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");
}

//Operazione di moltiplicazione
void moltiplica ()
{
	int  a,b = 0;
	printf ("Inserisci il primo numero da moltiplicare:\n");
	scanf ("%d", &a);
	printf ("Inserisci il secondo numero da moltiplicare:");
	scanf ("%d", &b);
	int prodotto = a * b;
```

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

//Operazione di moltiplicazione
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 ("Il risultato della divisione tra %d e %d e': %d", a,b,divisione);
}

//Inserimento Stringa
void ins_string ()
{
        char stringa[10];
    printf ("Inserisci la stringa:");
    scanf ("%{\0}s", &stringa);
}
```

PROGRAMMA CORRETTO

```c
// Assistente digitale


#include <stdio.h>

//Variabili Void
void menu ();
void moltiplica ();
void dividi ();
void ins_string();

//Casistiche per la scelta
int main ()

{
        char scelta = {'\0'};
        menu ();
        scanf ("%c", &scelta);
        switch (scelta)
        {
                case 'A':
                moltiplica();
                break;
                case 'B':
                dividi();
                break;
                case 'C':
                ins_string();
                break;
                default:
                printf ("Opzione non riconosciuta!");
                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");
}

//Operazione di moltiplicazione
void moltiplica ()
{
        int  a,b = 0;
        printf ("Inserisci il primo numero da moltiplicare:\n");
        scanf ("%d", &a);
             printf ("Inserisci il secondo numero da moltiplicare:");
        scanf ("%d", &b);
```

```c
        int divisione = a % b;

        printf ("Il risultato della divisione tra %d e %d e': %d", a,b,divisione);
}

//Inserimento Stringa
void ins_string ()
{
        char stringa[10];
        printf ("Inserisci la stringa:");
        scanf ("%{\0}s", &stringa);
}
```