

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



Esercizio_10_Epicode.c

Codice iniziale:

```
GNU nano 7.2
#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 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\n");
}
```

```

}

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 denominatore:");
    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);
}

```

1. Il codice presente nel file “Esercizio_10_Epicode.c”, è un programma che fornisce un menu all’utente, per effettuare alcune operazioni matematiche come: moltiplicazione, divisione e inserimento di una stringa. A seconda della scelta che farà l’utente, verrà chiamata una funzione che eseguirà l’operazione corrispondente.
2. Alcune casistiche non standard che il programma non gestisce:
 - L’utente può inserire un carattere diverso da “A”, “B” o “C”, ma il programma non gestisce questo problema;
 - Guardando la funzione “dividi” vediamo che, se l’utente inserisce un denominatore uguale a 0, il programma non riuscirà a gestirlo correttamente.
 - Nella funzione “moltiplica” legge in input due numeri come short int, dunque se l’utente digita un valore maggiore di 32767 (la massima rappresentazione short int), il programma non potrà gestire correttamente l’input.
- 3/4. Gli errori di sintassi/logici presenti nel codice da me identificati, sono:
 - Nella dichiarazione della variabile “scelta” in main(), è stato utilizzato un inizializzatore inutile con le parentesi graffe “{}”. La variabile dovrebbe essere dichiarata semplicemente come “char scelta = '\0';”

- Nel blocco “moltiplica()”, vengono usati gli specificatori di formato sbagliati per la lettura dei dati in input. La variabile “a” dovrebbe essere letta come “%d”, non “%f”, poiché si tratta di un intero. Questo potrebbe causare problemi se l’utente digita un valore con la virgola decimale.
Nello stesso blocco, la variabile b viene inizializzata a 0, il che significa che il secondo valore inserito dall’utente verrà sempre ignorato. Questa linea dovrebbe essere rimossa.
- Nel blocco “dividi()”, il calcolo della divisione utilizza l’operatore modulo “%” invece che l’operatore di divisione “/”. Questo restituirà il resto della divisione tra a e b anziché il risultato della divisione stessa. Dovrebbe essere corretto in “int divisione = a / b;”
- Infine, nel blocco “ins_string()”, quando si legge la stringa in input con “scanf()”, non è necessario utilizzare l’operatore “&” per ottenere l’indirizzo della variabile. scanf() prenderà automaticamente l’indirizzo della variabile. Quindi la riga corretta è “scanf(“%s”, stringa);” Per evitare il buffer overflow in questa funzione, utilizziamo fgets() invece di scanf().

Extra: Ho eseguito varie modifiche del codice per renderlo attivabile e senza bug:

```
(kali@kali)-[~/Desktop]
$ gcc EsModificato.c -o EsModificato

(kali@kali)-[~/Desktop]
$ ./EsModificato
Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti.
Come posso aiutarti?
A >> Moltiplicare due numeri.
B >> Dividere due numeri.
C >> Inserire una stringa.
█
```

```
(kali@kali)-[~/Desktop]
$ ./EsModificato
Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti.
Come posso aiutarti?
A >> Moltiplicare due numeri.
B >> Dividere due numeri.
C >> Inserire una stringa.
A
Inserisci i due numeri da moltiplicare: 1
4
Il prodotto tra 1 e 4 è: 4
```

```
(kali@kali)-[~/Desktop]
$ ./EsModificato
Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti.
Come posso aiutarti?
A >> Moltiplicare due numeri.
B >> Dividere due numeri.
C >> Inserire una stringa.
B
Inserisci il numeratore: 6
Inserisci il denominatore: 0
Impossibile dividere per zero. Riprova.
```

```
GNU nano 7.2 EsModificato.c *
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

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

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("Scelta non valida. Riprova.\n");
            break;
    }

    return 0;
}

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.\n");
}
```

```
GNU nano 7.2 EsModificato.c *
void moltiplica()
{
    int a, b;
    printf("Inserisci i due numeri da moltiplicare: ");
    scanf("%d %d", &a, &b);

    int prodotto = a * b;

    printf("Il prodotto tra %d e %d è: %d\n", a, b, prodotto);
}

void dividi()
{
    int a, b;
    printf("Inserisci il numeratore: ");
    scanf("%d", &a);

    printf("Inserisci il denominatore: ");
    scanf("%d", &b);

    if (b == 0)
    {
        printf("Impossibile dividere per zero. Riprova.\n");
        return;
    }

    int divisione = a / b;

    printf("La divisione tra %d e %d è: %d\n", a, b, divisione);
}

void ins_string()
{
    char stringa[100];
    printf("Inserisci la stringa: ");
    fgets(stringa, 100, stdin);

    // Rimuove il carattere newline dalla stringa
    stringa[strcspn(stringa, "\n")] = '\0';

    char risposta;
    printf("Vuoi stampare la stringa inserita? (S/N): ");
    scanf("%c", &risposta);

    if (risposta == 's' || risposta == 'S') {
        printf("La stringa inserita è: %s\n", stringa);
    }
}
```