

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

- Ad una prima analisi il programma propone una scelta all'utente di **A** moltiplicare e **B** dividere o **C** inserire una stringa. In caso si scelga **A moltiplica** si chiede di inserire due numeri interi **int** che verranno moltiplicati tra loro mostrandone il risultato; anche nel caso si scelga **B dividi** verranno chiesti due numeri interi che però verranno divisi, anche qui verrà mostrato il risultato ma usando l'operatore **%** viene calcolato del resto della divisione; infine, se si sceglie **C string** all'utente viene chiesto di inserire una stringa di massimo 9 caratteri (viene dichiarato un vettore di 10, ma i carattere che si possono inserire sono 9+1 che sarà il carattere di fine stringa).
- Nel codice sono presenti alcuni errori in quanto non viene prevista la divisione per 0, che da come risultato una divisione matematicamente impossibile, a mio avviso la stringa è troppo breve, ma potrebbe essere voluto, dovrebbe poi prevedere un errore in caso si superino i caratteri consentiti. Alcuni errori di sintassi sono poi segnalati nei righi corrispondenti con dei commenti al codice qui sottostante, ad esempio con la variabile **char**, che rappresenta un carattere, si dovrebbe usare il carattere **%c** (o al massimo **%s** per le stringhe di caratteri) invece di **%d** che è il carattere usato per indicare gli interi decimali. Poi, nella funzione moltiplica è presente il comando **short int** che è previsto nel linguaggio C++ e non in C, e i caratteri devono essere entrambi **d** cioè intero decimale con segno. Nella funzione divisione io userei l'operatore **/** che da il risultato della divisione

```
#include <stdio.h>
```

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

```
int main ()
```

```
{  
    char scelta = {'\0'}; //scritto male, si dovrebbe char scelta='\0' senza parentesi graffe.  
    menu ();  
    scanf ("%d", &scelta); //va messo c, invece di d  
  
    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; // il comando short è di C++ e non di C
    printf ("Inserisci i due numeri da moltiplicare:");
    scanf ("%f", &a); //no f ma %d
    scanf ("%d", &b);

    short int prodotto = a * b; // il comando short è di C++ e non di C
    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 denominator:");
    scanf ("%d", &b);

    int divisione = a % b; // inserirei la divisione in un if per segnalare l'errore matematico in caso di
divisione per 0(zero) perché impossibile.
    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);
}

```

Codice con correzioni:

```

#include <stdio.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("Opzione 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");
}

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

    if (b != 0)
    {
        int divisione = a / b;
        printf("La divisione tra %d e %d è: %d\n", a, b, divisione);
    }
    else
    {
        printf("Errore: Divisione per zero non consentita.\n");
    }
}

void ins_string()
{
    char stringa[10];
    printf("Inserisci la stringa: ");
    scanf("%9s", stringa);
}

```