

Traccia: Esercizio Osservazione critica 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.

Codice corretto:

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

```
void menu();
```

```
void moltiplica();
```

```
void dividi();
```

```
void ins_string();
```

```
int main() {
```

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

```
    menu();
```

```
    scanf(" %c", &scelta); //Ho aggiunto uno spazio prima di %c in scanf per ignorare eventuali spazi bianchi
```

```
    switch (scelta) { //Ho aggiunto i case per le lettere minuscole ('a', 'b', 'c') per rendere il programma case-insensitive, cioè accetta sia maiuscole che minuscole per le scelte.
```

```
        case 'A':
```

```
        case 'a':
```

```
            moltiplica();
```

```
            break;
```

```
        case 'B':
```

```
        case 'b':
```

```
            dividi();
```

```
            break;
```

```
        case 'C':
```

```

    case 'c':
        ins_string();
        break;
    default:
        printf("Scelta non valida.\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() { //Ho corretto i tipi di dati per la funzione moltiplica per essere int invece di
short int.

//Ho modificato scanf per leggere entrambi i numeri in una singola chiamata, per renderlo più
conciso e pulito.

    int a = 0, 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 = 0, b = 0;
```

```
    printf("Inserisci il numeratore: ");
```

```
    scanf("%d", &a);
```

```
    printf("Inserisci il denominatore: ");
```

```
    scanf("%d", &b);
```

if (b != 0) { //Ho aggiunto un controllo per evitare la divisione per zero, che altrimenti causerebbe un errore. Ho cambiato l'operatore da % a / e ho castato a a float per assicurare che la divisione sia in virgola mobile e non una divisione intera.

```
    float divisione = (float) a / b;
```

```
    printf("La divisione tra %d e %d è: %.2f\n", a, b, divisione);
```

```
    } else {
```

```
        printf("Errore: divisione per zero non consentita.\n");
```

```
    }
```

```
}
```

void ins\_string() { //Ho aumentato la dimensione dell'array per la stringa a 100 caratteri per evitare overflow.

```
    char stringa[100];
```

```
    printf("Inserisci la stringa: ");
```

```
    scanf("%s", stringa);
```

```
    printf("Hai inserito: %s\n", stringa);
```

```
}
```