

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

Il programma sembra essere un assistente digitale che presenta un menu all'utente con tre opzioni (A, B, C):

A.Moltiplicare due numeri

B.Dividere due numeri

C.Inserire una stringa

L'utente può scegliere un'opzione inserendo la lettera corrispondente, dopodiché verrà eseguita la funzione associata all'opzione scelta.

-Individuare eventuali errori di sintassi / logici

ERRORI LOGICI

-Alla riga 14-> %d è sbagliato perchè indica che ci si aspetta di acquisire un numero intero ma in questo caso la variabile scelta è di tipo char.

```
14 scanf ("%d", &scelta);
```

```
20 scanf (" %c", &scelta);
```

-Nel "void moltiplica" iniziale sono presenti sia %f che %d con "short int" e questo non è corretto. Ho deciso di trattare a e b come numeri interi, precisamente come short int (%hd).

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

    short int prodotto = a * b;

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

-Per dividere due numeri è necessario l'operatore / e non % (riga 64) ->

```
64      int divisione = a % b;
```

```
77      float divisione = a / b;
```

-Nella divisione -> ho deciso di utilizzare float (numeri reali - %f) perchè capita spesso che il risultato della divisione non sia un numero intero.

```
55
56 void dividi ()
57 {
58     int a,b = 0;
59     printf ("Inserisci il numeratore:");
60     scanf ("%d", &a);
61     printf ("Inserisci il denominatore:");
62     scanf ("%d", &b);
63
64     int divisione = a % b;
65
66     printf ("La divisione tra %d e %d e': %d", a,b,divisione);
67 }
68
69
```

%.2f: vuol dire che verranno visualizzati 2 numeri dopo la virgola.

```
68
69 void dividi ()
70 {
71     float a,b = 0;
72     printf ("Inserisci il numeratore:");
73     scanf (" %f", &a);
74     printf ("Inserisci il denominatore:");
75     scanf (" %f", &b);
76
77     float divisione = a / b;
78
79     printf ("La divisione tra %.2f e %.2f e': %.2f", a,b,divisione);
80 }
81
82
```

-Nell'utilizzo di scanf per leggere una stringa il formato di lettura è corretto (%s) ma non è necessario aggiungere l'operatore & prima di "stringa".

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

```
91      scanf (" %19s", stringa);
```

ERRORI DI SINTASSI

-Alla riga 12 -> ho eliminato le parentesi graffe perchè non erano necessarie. Le parentesi graffe si utilizzano quando si apre una nuova funzione.

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

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

-Alla riga 36 c'è un errore ortografico -> Ho aggiunto la "s" mancante.

```
36      printf ("Benvenuto, sono un assistente digitale,
```

```
49      printf ("Benvenuto, sono un assistente digitale,
```

-C'è un altro errore ortografico alla riga 61 ->

```
61      printf ("Inserisci il denumeratore:");
```

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

- Individuare dal codice sorgente le casistiche non standard che il programma non gestisce (esempio, comportamenti potenziali che non sono stati contemplati)
- Proporre una soluzione per ognuno di essi

-Il programma non dava la possibilità di inserire le risposte anche con le lettere minuscole. Ho quindi implementato nello switch i tre case relativi ad "a,b,c".

```
9  int main ()
10 {
11     char scelta = {'\0'};
12     menu ();
13     scanf ("%d", &scelta);
14
15     switch (scelta)
16     {
17         case 'A':
18             multiplica();
19             break;
20         case 'B':
21             dividi();
22             break;
23         case 'C':
24             ins_string();
25             break;
26     }
27
28     return 0;
29 }
30
31
```

```
16  int main ()
17  {
18      char scelta = '\0';
19      menu ();
20      scanf (" %c", &scelta);
21
22      switch (scelta)
23      {
24          case 'A':
25              multiplica();
26              break;
27          case 'B':
28              dividi();
29              break;
30          case 'C':
31              ins_string();
32              break;
33          case 'a':
34              multiplica();
35              break;
36          case 'b':
37              dividi();
38              break;
39          case 'c':
40              ins_string();
41              break;
42      }
43      return 0;
44  }
45
```

-Ho aggiunto "pulisciBuffer" sia per un motivo estetico che per "pulire" il buffer di input da eventuali caratteri residui, inclusi i caratteri di nuova linea, che altrimenti potrebbero influenzare la successiva acquisizione dell'input utente. Quindi questa funzione permette di evitare che il programma vada in loop.

```
1  #include <stdio.h>
2
3  void menu ();
4  void moltiplica ();
5  void dividi ();
6  void ins_string();
7
8
9  int main ()
10
```

```
1  #include <stdio.h>
2  #include <stdbool.h>
3  #include <string.h>
4
5  void pulisciBuffer() {
6      int c;
7      while ((c = getchar()) != '\n' && c != EOF);
8  }
9
10 void menu ();
11 void moltiplica ();
12 void dividi ();
13 void ins_string();
14
15
```

-Inizialmente il "void ins_string" permetteva solo all'utente di inserire una qualsiasi stringa e non era neanche corretto. Ho deciso di rifarlo completamente in modo tale da considerare anche la lunghezza della stringa e riconoscere come valide solo le stringhe inserite con massimo 18 caratteri.

```
72
73 void ins_string ()
74 {
75     char stringa[10];
76     printf ("Inserisci la stringa:");
77     scanf ("%s", &stringa);
78 }
79
```

```
85
86 void ins_string () {
87     bool lunghezzastringa = true;
88     char stringa[20];
89     do {
90         printf ("Inserisci la stringa:");
91         scanf ("%19s", stringa);
92         if (strlen(stringa) > 18) {
93             printf("Non valido. Nome troppo lungo\n");
94             pulisciBuffer();
95             lunghezzastringa = true; //per far ripartire il ciclo di questo do-while
96         } else {
97             printf("La tua stringa è: %s", stringa);
98             lunghezzastringa = false;
99         }
100     } while (lunghezzastringa);
101 }
102
```