Traccia S2L5:

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.

Il programma di oggi è stato scritto usando il linguaggio C. Osservando il programma possiamo dedurre che manderà a schermo un menù dove l'utente può scegliere se eseguire una moltiplicazione,una divisione oppure inserire una stringa; però sono presenti alcuni errori.

A seguire una lista degli errori che ho rilevato,
alla fine ci saranno anche le correzioni che ho apportato:

```
11 {
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          }
```

Errori dentro la funzione int main():

Riga 14: **scanf ("%d", &scelta)** c'è un errore di tipo sintattico, non stiamo andando a leggere un **int**, ma un **char**.

Da Riga 16 a 26: Nello Switch non viene tenuto conto della casistica di **default** inoltre non vengono tenuti conto dei caratteri in minuscolo **"a","b","c"** che potrebbero essere inseriti in input.
Questo è un errore logico.

```
43 void moltiplica ()
44 {
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);
```

Errori all'interno della funzione moltiplica():

Riga 45 e 50: l'utilizzo di **short int** in una moltiplicazione fra numeri non è la scelta più adatta poiché **short int** è un range limitato (un minimo di -32,768 e un massimo di 32,767). Meglio usare **int**.
Questo è un errore di tipo logico.

Riga 47 e 48: l'utilizzo di **"%f"** sta indicare che noi ci aspettiamo un valore float mentre **"%d"** il valore di tipo int e in questo caso usando **short int** dovrebbe essere corretto "**%hd**".
Questo è un errore di tipo sintattico.

```
56 void dividi ()
57 {
58         int  a,b = 0;
59         printf ("Inserisci il numeratore:");
60         scanf ("%d", &a);
61         printf ("Inserisci il denumeratore:");
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 }
```

Errori presenti all'interno della funzione dividi():

Riga 58 e 64: l'inserimento di **int** all'interno di una divisione non è consigliata poiché il risultato potrebbe contenere un numero non intero. **Meglio usare float.**
Errore di tipo logico.

Riga 60 e 62: l'utilizzo di **"%d"** di conseguenza non è consono perché dal risultato ci aspettiamo un valore di tipo **float**. Errore di tipo logico.

Riga 64: il risultato sarà di tipo **float** e non **int**, inoltre l'espressione scritta restituirà il **modulo** dei due numeri e non la **divisione**. Errore di tipo Logico

Riga 66: di conseguenza i **"%d"** inseriti qua non sono corretti e dovranno essere sostituiti con **"%f"**. Questo è un errore di tipo Logico.

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

Errori presenti all'interno della funzione ins_string():

Riga 76: Non è presente un controllo sulla lunghezza dell'input, ciò mette a rischio il programma esponendolo a un **possibile attacco overflow**. Errore di tipo Logico.

Inoltre la presenza della **"&"** non è necessaria dato che il puntatore già è indirizzato verso la prima posizione dell'array, infine questa parte di codice non restituirà a schermo la stringa inserita in input.

Fine di analisi del codice.

Tengo a precisare che ci sono anche alcuni errori di battitura che ho corretto. Successivamente ho modificato la parte dello **switch** inserendo **"a", "b" e "c"** come caratteri accettati e aggiunto un caso di **default** in caso di input errato. Ho cambiato la parte della **moltiplicazione** seguendo le notazioni sopra riportate. Ho riscritto la parte della **divisione** inserendo un ciclo **while** per evitare che l'utente inserisca la **divisione per 0**. Ho aggiunto un controllo nella parte dell'inserimento della stringa per evitare che **superi i caratteri richiesti** e la possibilità di **visualizzare a schermo** la stringa inserita.

Codice in Analisi:

```
 9 int main ()
10
11 {
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         }
```

Codice con correzioni:

```
11 {
12         char scelta ={'\0'};
13         menu ();
14         scanf ("%c", &scelta);
15
16         switch (scelta) {
17                 case 'A':
18                 case 'a':
19                  moltiplica();
20                 break;
21
22                 case 'B':
23                 case 'b':
24                 dividi();
25
26                 break;
27                 case 'C':
28                 case 'c':
29
30                 ins_string();
31                 break;
32
33                 default:
34                 printf("Scelta non valida.\n");
35                 break;
36 }
37
```

```
43 void moltiplica ()
44 {
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);
53 }
```

```
52 void moltiplica ()
53 {
54         int  a,b = 0;
55         printf ("Inserisci i due numeri da moltiplicare:");
56         scanf ("%d", &a);
57         scanf ("%d", &b);
58
59         int prodotto = a * b;
60
61         printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto);
62 }
```

```
56 void dividi ()
57 {
58         int  a,b = 0;
59         printf ("Inserisci il numeratore:");
60         scanf ("%d", &a);
61         printf ("Inserisci il denumeratore:");
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 }
```

```
65 void dividi ()
66 {
67         float  a,b = 0;
68         printf ("Inserisci il numeratore:");
69         scanf ("%f", &a);
70         while (b == 0)
71         {
72         printf ("Inserisci il denominatore diverso da 0:");
73         scanf ("%f", &b);
74         }
75
76         float divisione = a / b;
77
78         printf ("La divisione tra %.2f e %.2f e': %.2f\n", a,b,divisione);
79 }
```

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

```
85 void ins_string() {
86     char stringa[10];
87     printf("Inserisci la stringa (massimo 9 caratteri): ");
88     scanf("%9s", stringa);
89
90     printf("Hai inserito: %s\n", stringa);
91 }
```