

PROGETTO S2/L5

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.

FUNZIONALITÀ DEL CODICE

Il programma in questione è un assistente digitale che permette di effettuare moltiplicazioni, divisioni o inserire una stringa.

ERRORI NEL CODICE

ERRORE 1 – Errore di sintassi

```
12      char scelta = {'\0'};  
13      menu ();  
14      scanf ("%d", &scelta);  
15
```

Dato che si è dichiarata la variabile "Char", l'indicatore preciso da inserire nello scanf è %c, relativo a quella variabile.

ERRORE 2 – Errore di sintassi

```
12      char scelta = {'\0'};  
13      menu ();  
14      scanf ("%d", &scelta);  
15
```

Stessa parte di codice di prima, nello scanf %d che abbiamo chiarito sia %c, è preferibile mettere una spazio tra la prima doppia virgoletta e la %c. Questo permette di leggere meglio il carattere desiderato, assicurando che l'input venga interpretato correttamente

ERRORE 3 – Errore di casistica non gestita

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

In questo blocco se l'utente non inserisce A, B e C in maiuscolo avrà problemi con il programma, quindi è giusto inserire la casistica in cui si aggiunga anche a, b e c minuscolo.

ERRORE 4 – Errore di casistica non gestita

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

Manca l'inserimento del Default all'interno della funzione Switch. Non è inserito dunque la gestione dei casi diversi dalle tre opzioni selezionabili, in modo tale che se l'input dell'utente non è valido, gli permetta di ripetere l'inserimento.

ERRORE 5 – Errore di sintassi

```
45     short int a,b = 0;
46     printf ("Inserisci i due numeri da moltiplicare:");
47     scanf ("%f", &a);
48     scanf ("%d", &b);
49
```

Qui bisogna fare attenzione alla variabile di tipo short int, perchè ha un valore di 2 byte quindi ha un range minore rispetto a variabili di tipo int. È preferibile inserire quindi int al posto di short int. Anche a scanf(%f...) si sostituirà con %d.

ERRORE 6 – Errore di sintassi

```
short int prodotto = a * b;

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

Anche in questo caso il tipo di variabile giusto da mettere è int.

ERRORE 7 – Errore di sintassi

```
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 }
```

In questo blocco di codice è meglio mettere la variabile di tipo float invece di int perchè il risultato può non essere sempre un numero intero.

ERRORE 8 – Errore di sintassi

```
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 }
```

L'errore è mettere l'operatore "%" come divisione.

ERRORE 9 – Errore di casistica non gestita

```
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 }
```

Sempre in questo blocco di codice bisogna specificare la casistica in cui il denominatore inserito sia 0. Questo creerebbe un problema al programma.

ERRORE 10 – Errore di sintassi

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

In questo blocco di codice la parte di stringa punta già ad un indirizzo di memoria essendo un array, quindi si può evitare di inserire la &.

ERRORE 11 – Errore di casistica non gestita

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

Sempre in questo blocco è un altro errore importante, non si dà un limite di carattere da inserire all'utente. Se quest'ultimo inserisce più di 10 caratteri il programma va in Stack Overflow.

SOLUZIONE DEL CODICE

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

SOLUZIONE SINTASSI RIGA 14: andiamo a correggere l'errore andando ad inserire "scanf(" %c". &scelta);", perchè la variabile dichiarata è char. Andiamo ad inserire anche uno spazio tra la virgoletta e % per una lettura migliore del carattere dal parte del programma e per assicurarci che l'imput venga eseguito correttamente.

SOLUZIONE CASISTICA NON GESTITA RIGHE 18/22/26: andiamo ad inserire nella riga 19/23/27 la casistica in cui l'utente preme le lettere scelte in minuscolo.

SOLUZIONE CASISTICA NON GESTITA RIGA 30: andiamo ad inserire la funzione default nel caso in cui l'utente digiti una lettera diversa da "a, b o c". Uscirà scritta che la scelta non è valida e ritornerà all'inserimento della scelta.

```
49 void multiplica ()
50 {
51     int a,b = 0;
52     printf ("Inserisci i due numeri da moltiplicare:");
53     scanf ("%d", &a);
54     scanf ("%d", &b);
55
56     int prodotto = a * b;
57
58     printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto);
59 }
60
```

SOLUZIONE RIGHE 51: andiamo ad inserire int al posto di short int per il basso numero byte di quest'ultimo. Così facendo si potranno inserire un maggior range di numeri.

SOLUZIONE RIGHE 53: si va a sostituire %f con %d in quando si è scelto int.

SOLUZIONE RIGA 56: inseriamo anche qui int invece di shot int.

```
62 void dividi ()
63 {
64     float a,b = 0;
65     printf ("Inserisci il numeratore:");
66     scanf ("%f", &a);
67     while(b == 0)
68     {
69         printf ("Inserisci il denominatore:");
70         scanf ("%f", &b);
71     }
72     float divisione = a / b;
73
74     printf ("La divisione tra %f e %f e': %f", a,b,divisione);
75 }
76
```

SOLUZIONE RIGA 64: andiamo a sostituire int con float perchè il risultato può non essere sempre intero.

SOLUZIONE RIGHE 66 e 70: in queste due righe sostituiamo %d con %f sempre per la sostituzione precedente da int a float.

SOLUZIONE RIGA 67: andiamo ad inserire la funzione while nel caso in cui l'utente metta come denominatore 0. Questa farà inserire il numero 0 fin quando non si cambierà in un altro numero.

SOLUZIONE RIGA 72: anche qui andiamo a mettere float invece che int. Inoltre si è andato a cambiare l'operatore % con l'operatore /.

SOLUZIONE RIGA 74: anche qui andiamo a sostituire %d con %f.

```
78
79 void ins_string ()
80 {
81     char stringa[10];
82     printf ("Inserisci la stringa:");
83     scanf ("%9s", stringa);
84 }
85
```

SOLUZIONE RIGA 83: si è aggiunto il valore 9 in %s per dare un limite al carattere inseribile dall'utente per non creare stack overflow. Inoltre si è tolta la & perchè inserendo già la stringa di char va a già a puntare ad un indirizzo di memoria.