

S2L5

Programmazione di un assistente digitale che può compiere le seguenti funzioni

- Moltiplicare 2 numeri
- Dividere 2 numeri
- Inserire una stringa

```
File Edit Search View Document Help
1
2 #include <stdio.h>
3
4 void menu ();
5 void moltiplica ();
6 void dividi ();
7 void ins_string();
8
9 int main ()
10 {
11     char scelta = {'\0'};
12     menu ();
13     scanf ("%d", &scelta);
14
15     switch (scelta)
16     {
17         case 'A':
18             moltiplica();
19             break;
20         case 'B':
21             dividi();
22             break;
23         case 'C':
24             ins_string();
25             break;
26         default:
27             return 0;
28     }
29 }
```

(Riga 4; 5 e 6) Ho eliminato lo spazio bianco prima delle parentesi per convenzione ai lavori in Team o con altri linguaggi di programmazione cioè migliora la leggibilità.

(Riga 11) Quel valore si usa per indicare la fine di una catena de caratteri in questo caso non c'è necessità di usarlo perché è una variabile di tipo char. Eliminando {'\0'} il codice diventa più pratico e semplice.

(Riga 13) "%d" è sbagliato perché quel definitore si usa solo per interi. In questo caso il definitore corretto è "%c". Poi ho aggiunto uno spazio bianco per evitare problemi con caratteri in altre linee.

(Riga 17; 20 e 23) Qui ho aggiunto 2 casi per ogni lettera (A,B o C) nello Switch, per rendere il programma insensibile al uso di maiuscole e minuscole, migliorando così l'UX.

(Riga 26) Inserimento **default**: in caso nessuna delle opzioni corrisponda al valore della variabile il programma stampa un messaggio che indica " Opzione non valida" e si chiude.

Importanza di usare default:

L'attaccante potrebbe iniettare malware sfruttando queste vulnerabilità.

```

30 void menu (){
31     printf ("Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti\n");
32     printf ("Come posso aiutarti?\n");
33     printf ("A >> Moltiplicare due numeri\nB >> Dividere due numeri\nC >> Inserire una stringa\n");
34 }
35 void moltiplica (){
36     short int a,b = 0;
37     printf ("Inserisci i due numeri da moltiplicare:");
38     scanf ("%f", &a);
39     scanf ("%d", &b);
40
41     short int prodotto = a * b;
42
43     printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto);
44 }

```

(Riga 38 e 39) Prima ho cambiato la variabile «short int» e ho messo «int» perché l'intervallo di numeri è più grande rispetto a «short» e mi facilita la moltiplicazione ampliando la limitazione

(Riga 38 e 39) Ho cambiato i definitori corretti che sarebbero «%d» (perché le variabili sono interi). Poi ho aggiunto due condizionali «if» in caso l'utente inserisca caratteri non numerici («!= 1») in questo caso se il valore inserito da «scanf» è diverso da 1 (int) non potrà assegnare un valore a la variabile «a» e stamperà un messaggio di return «inserire solo carattere numerico»

Importanza di usare i definitori/variabili/operatori corretti:

L'attaccante potrebbe tentare di attaccare la stringa rubando informazioni, potrebbe inserire malware per ottenere accessi non autorizzati.

```

46 void dividi ()
47 {
48     int a,b = 0;
49     printf ("Inserisci il numeratore:");
50     scanf ("%d", &a);
51     printf ("Inserisci il denominatore:");
52     scanf ("%d", &b);
53
54     int divisione = a % b;
55
56     printf ("La divisione tra %d e %d e': %d", a,b,divisione);
57 }

```

(Riga 48 e 54) ho cambiato alla variabile «float» perché nelle divisioni non sempre vengono numeri interi di conseguenza ho cambiato anche i definitori.

(Riga 56) In questa parte ho aggiunto una condizionale per controllare che b (denominatore) sia diverso da 0 «(b != 0)» perché la divisione per zero è impossibile. **Se** b non è 0 viene inizializzata la variabile «divisione» con il risultato della divisione (sempre numeri int). **Invece** se b è 0 viene eseguito «else» e si stampa un messaggio che indica che è impossibile

(Riga 50 e 52) Qui, nello stesso modo della funzione moltiplica, ho aggiunto due condizionali «if» in caso l'utente inserisce caratteri non numerici («!= 1»)

(Riga 54) Operatore sbagliato.
Per la divisione si usa «/»

Importanza dell'uso del if-else:

L'attaccante potrebbe tentare di sfruttare le vulnerabilità di controllo del flusso per mancanza di convalida dell'input.

```
58
59 void ins_string ()
60 {
61     char stringa[10];
62     printf ("Inserisci la stringa:");
63     scanf ("%s", &stringa);
64 }
```

(Riga 63) Qui dopo «scanf» ho specificato «%9s» perché indica a «scanf» di leggere al massimo 9 caratteri (per evitare overflow dell'array perché il suo limite massimo è di 9) per poi memorizzarli nell'array. In fine ho aggiunto un «printf» per stampare la stringa inserita utilizzando il definitore «%s»

(Riga 61) ho inserito «\0» per inizializzare l'array con 0 per capire che stiamo creando un array di 10 caratteri in cui ogni carattere è impostato su '\0'. In C è utile chiarire questo per poter lavorare con altre funzioni e librerie.

Importanza di inizializzare l'array:

L'attaccante potrebbe tentare di eseguire attacchi overflow inserendo dati fuori dal limite

CODICE CORRETTO SU KALI:

```
1 #include <stdio.h>
2
3 void menu();
4 void moltiplica();
5 void dividi();
6 void ins_string();
7
8 int main() {
9     char scelta;
10    menu();
11    scanf(" %c", &scelta);
12    switch (scelta) {
13        case 'a':
14        case 'A':
15            moltiplica();
16            break;
17        case 'b':
18        case 'B':
19            dividi();
20            break;
21        case 'c':
22        case 'C':
23            ins_string();
24            break;
25        default:
26            printf("Opzione non valida.\n");
27    }
28    return 0;
29 }
```

```

30
31 void menu() {
32     printf("Benvenuto, sono un assistente digitale, posso
    aiutarti a sbrigare alcuni compiti\n");
33     printf("Come posso aiutarti?\n");
34     printf("a >> Moltiplicare due numeri\nb >> Dividere due
    numeri\nc >> Inserire una stringa\n");
35 }
36
37 void moltiplica() {
38     int a, b=0;
39     printf("Inserisci i due numeri da moltiplicare:\n");
40     if (scanf("%d", &a) != 1) {
41         printf("Inserire solo carattere numerico.\n");
42         return;
43     }
44     if (scanf("%d", &b) != 1) {
45         printf("Inserire solo carattere numerico.\n");
46         return;
47     }
48     int prodotto = a * b;
49     printf("Il prodotto tra %d e %d è: %d\n", a, b, prodotto);
50     }
51
52 void dividi() {
53     float a, b=0;
54     printf("Inserisci il numeratore:");
55     if (scanf("%f", &a) != 1) {
56         printf("Inserire solo carattere numerico.\n");
57         return;
58     }
59     printf("Inserisci il denominatore:");
60     if (scanf("%f", &b) != 1) {
61         printf("Inserire solo carattere numerico.\n");
62         return;
63     }
64     if (b != 0) {
65         float divisione = a / b;
66         printf("La divisione tra %.2f e %.2f è: %.2f\n", a, b,
    divisione);
67     } else { printf("Impossibile dividere per
    zero.\n");
68     }
69     }
70
71 void ins_string() {
72     char stringa[10]={'\0'};
73     printf("Inserisci la stringa:");
74     scanf("%9s", stringa);
75     printf("La stringa inserita è: %s\n", stringa);
76     }

```

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

```
void menu();
```

```
void moltiplica();
```

```
void dividi();
```

```
void ins_string();
```

```
int main() {
```

```
    char scelta;
```

```
    menu();
```

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

```
    switch (scelta) {
```

```
        case 'a':
```

```
        case 'A':
```

```
            moltiplica();
```

```
            break;
```

```
        case 'b':
```

```
        case 'B':
```

```
            dividi();
```

```
            break;
```

```
        case 'c':
```

```
        case 'C':
```

```
            ins_string();
```

```
            break;
```

```
        default:
```

```
            printf("Opzione non valida.\n");
```

```
    }
```

```
    return 0;
```

```
}
```

```
void menu() {
```

```
    printf("Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti\n");
```

```
    printf("Come posso aiutarti?\n");
```

```
    printf("a >> Moltipicare due numeri\nb >> Dividere due numeri\nc >> Inserire una stringa\n");
```

```
}
```

```
void moltiplica() {
```

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

```

printf("Inserisci i due numeri da moltiplicare:\n");

if (scanf("%d", &a) != 1) {

    printf("Inserire solo carattere numerico.\n");

    return;

}

if (scanf("%d", &b) != 1) {

    printf("Inserire solo carattere numerico.\n");

    return;

}

int prodotto = a * b;

printf("Il prodotto tra %d e %d è: %d\n", a, b, prodotto);

    }

```

```

void dividi() {

    float a, b=0;

    printf("Inserisci il numeratore:");

    if (scanf("%f", &a) != 1) {

        printf("Inserire solo carattere numerico.\n");

        return;

        }

    printf("Inserisci il denominatore:");

    if (scanf("%f", &b) != 1) {

        printf("Inserire solo carattere numerico.\n");

        return;

        }

    if (b != 0) {

        float divisione = a / b;

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

        } else { printf("Impossibile dividere per zero.\n");

        }

    }
}

```

```

void ins_string() {

    char stringa[10]={'\0'};

    printf("Inserisci la stringa:");

    scanf("%9s", stringa);

    printf("La stringa inserita è: %s\n", stringa);

    }
}

```