

# ESERCITAZIONE W6D1(2)



## Esercizio

Ottimizzazione del codice

### Traccia:

Ieri abbiamo visto come scrivere un piccolo gioco di domande e risposte in C.

Oggi pensiamo all'ottimizzazione del codice, ed alla gestione delle situazioni non previste.

Riprendete il codice del programma che avete scritto ieri e facciamo le seguenti considerazioni:

- Cosa succede se l'utente inserisce una lettera **diversa** da A o B in fase di scelta iniziale? Il programma termina, ma non è una casistica che abbiamo gestito.
- Cosa succede se l'utente inserisce un nome che ha più caratteri della dimensione dell'array «nome» che abbiamo dichiarato inizialmente nella fase di avvio nuova partita? Riceveremo un errore (provate ad inserire una sequenza molto lunga di caratteri)
- Cosa succede se l'utente inserisce la lettera D per la risposta alle domande durante una partita? O un carattere numerico?

Tutte queste situazioni vanno considerate in fase di programmazione in quanto errori logici o errori di mancata gestione di situazioni non standard potrebbero portare a bug nel codice che potrebbero essere sfruttati da un attaccante per prendere controllo dell'esecuzione del programma ed eseguire codice malevolo.

Per gestire queste casistiche, possiamo apportare alcune modifiche al codice:

1. Gestione della scelta iniziale: Aggiungiamo un loop che continua fino a quando l'utente non inserisce una scelta valida (A o B).
2. Gestione del nome: Possiamo modificare la scansione dell'input per leggere una stringa fino a un massimo di caratteri, in modo da evitare overflow nell'array «nome».
3. Gestione delle risposte durante il gioco: Aggiungiamo un controllo per verificare se l'utente ha inserito una risposta valida (1, 2, 3 o 4).

Ecco il codice aggiornato con queste modifiche:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  #define MAX_DOMANDE 3
6  #define MAX_CARATTERI_NOME 50
7
8  typedef struct {
9      char domanda[100];
10     char risposte[4][50];
11     int risposta_corretta;
12 } Domanda;
13
14 void gioca() {
15     Domanda domande[MAX_DOMANDE] = {
16         {"Qual è la capitale dell'Italia?",
17          {"A) Roma\n", "B) Milano\n", "C) Napoli\n", "D) Firenze\n"},
18          1},
19         {"Qual è l'organo più grande del corpo umano?",
20          {"A) Cuore\n", "B) Fegato\n", "C) Pelle\n", "D) Polmoni\n"},
21          3},
22         {"Chi ha scritto \"Il Signore degli Anelli\"?",
23          {"A) J.K. Rowling\n", "B) George Orwell\n", "C) J.R.R. Tolkien\n", "D) C.S. Lewis\n"},
24          3}
25     };
26
27     char nome[MAX_CARATTERI_NOME];
28     printf("\nInserisci il tuo nome: ");
29     scanf("%49s", nome); // Legge fino a 49 caratteri per evitare buffer overflow
30
31     int punteggio = 0;
32     for (int i = 0; i < MAX_DOMANDE; i++) {
33         printf("\nDomanda %d: %s", i + 1, domande[i].domanda);
34         for (int j = 0; j < 4; j++) {
```

```

34     for (int j = 0; j < 4; j++) {
35         printf("%s", domande[i].risposte[j]);
36     }
37     int risposta;
38     printf("Inserisci la tua risposta (1-4): ");
39     while (scanf("%d", &risposta) != 1 || risposta < 1 || risposta > 4) {
40         printf("Risposta non valida. Inserisci un numero da 1 a 4: ");
41         scanf("%*[^\\n]"); // Pulisce il buffer di input
42     }
43     if (risposta == domande[i].risposta_corretta) {
44         printf("Risposta corretta!\\n");
45         punteggio++;
46     } else {
47         printf("Risposta sbagliata. La risposta corretta era la %d.\\n", domande[i]
48             .risposta_corretta);
49     }
50     printf("\\n%s, hai totalizzato %d punti.\\n", nome, punteggio);
51 }
52
53 int main() {
54     char scelta;
55     do {
56         printf("\\n\\nBenvenuto al gioco di domanda/risposta!\\n");
57         printf("Scegli cosa vuoi fare:\\n");
58         printf("A) Inizia una nuova partita\\n");
59         printf("B) Esci\\n");
60         printf("Scelta: ");
61         while (scanf(" %c", &scelta) != 1 || (scelta != 'A' && scelta != 'a' && scelta != 'B' &&
62             scelta != 'b')) {
63             printf("Scelta non valida. Riprova: ");
64             scanf("%*[^\\n]"); // Pulisce il buffer di input
65         }
66         switch (scelta) {

```

```

66             case 'A':
67             case 'a':
68                 gioca();
69                 break;
70             case 'B':
71             case 'b':
72                 printf("\\nArrivederci!\\n");
73                 break;
74         }
75     } while (scelta != 'B' && scelta != 'b');
76
77     return 0;
78 }
79

```