

Parte 2

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <ctype.h>
5
6  // Struttura per rappresentare una domanda
7  typedef struct {
8      char testo[100];
9      char risposte[3][50];
10     int rispostaCorretta;
11 } Domanda;
12
13 // Funzione per mostrare il menu iniziale e ottenere la scelta dell'utente
14 char mostraMenuIniziale() {
15     char scelta;
16     do {
17         printf("Benvenuto al Gioco di Domanda/Risposta!\n");
18         printf("A) Iniziare una nuova partita\n");
19         printf("B) Uscire dal gioco\n");
20         printf("Scegli: ");
21         scanf(" %c", &scelta);
22
23         // Controllo se l'input è valido (A o B)
24         if (toupper(scelta) != 'A' && toupper(scelta) != 'B') {
25             printf("Scelta non valida. Inserisci A o B.\n");
26         }
27     } while (toupper(scelta) != 'A' && toupper(scelta) != 'B');
28
29     return toupper(scelta);
30 }
```

Gestione dell'Input Iniziale:

Nella funzione `mostraMenuIniziale`, ho implementato una gestione più robusta dell'input iniziale. Il loop continua fino a quando l'utente fornisce un input corretto (A o B), e la conversione in maiuscolo con `toupper` aiuta a garantire la correttezza anche se vengono inserite le minuscole.

```

32 // Funzione per gestire una nuova partita
33 void nuovaPartita() {
34     int punteggio = 0;
35     char nome[50];
36
37     // Ottieni il nome del giocatore in modo sicuro
38     // Dentro la funzione nuovaPartita():
39     do {
40         printf("Inserisci il tuo nome (massimo 49 caratteri): ");
41         fgets(nome, sizeof(nome), stdin);
42
43         // Rimuovi il carattere di nuova linea finale, se presente
44         size_t lunghezzaNome = strlen(nome);
45         if (lunghezzaNome > 0 && nome[lunghezzaNome - 1] == '\n') {
46             nome[lunghezzaNome - 1] = '\0';
47         }
48
49         // Controlla se il nome è troppo lungo
50         if (strlen(nome) >= 49) {
51             printf("Nome troppo lungo. Riprova.\n");
52         }
53     } while (strlen(nome) >= 49);
54
55

```

Gestione della Lunghezza del Nome:

Nella fase di acquisizione del nome all'interno della funzione `nuovaPartita`, ho implementato una verifica sulla lunghezza del nome. L'utilizzo di `fgets` aiuta a prevenire potenziali overflow del buffer, garantendo la sicurezza durante la lettura della stringa.

```

// Array di domande
Domanda domande[5] = {
    {"Qual è la capitale dell'Italia?", {"Roma", "Parigi", "Berlino"}, 0},
    {"Quanto fa 2 + 2?", {"3", "4", "5"}, 1},
    {"Chi è l'autore di Romeo e Giulietta?", {"Shakespeare", "Dante", "Hemingway"}, 0},
    {"Quale pianeta è conosciuto come la 'Sorella della Terra'?", {"Marte", "Venere", "Giove"}, 1},
    {"Quante corde ha una chitarra classica?", {"4", "6", "8"}, 1}
};

// Loop attraverso le domande
for (int i = 0; i < 5; i++) {
    printf("\nDomanda %d: %s\n", i + 1, domande[i].testo);
    printf("A) %s\nB) %s\nC) %s\n", domande[i].risposte[0], domande[i].risposte[1], domande[i].risposte[2]);

    // Controllo sicuro dell'input della risposta
    do {
        printf("Inserisci la tua risposta (A, B o C): ");
        fgets(rispostaUtente, sizeof(rispostaUtente), stdin);

        // Rimuovi il carattere di nuova linea finale, se presente
        size_t lunghezzaRisposta = strlen(rispostaUtente);
        if (lunghezzaRisposta > 0 && rispostaUtente[lunghezzaRisposta - 1] == '\n') {
            rispostaUtente[lunghezzaRisposta - 1] = '\0';
        }

        if (toupper(rispostaUtente[0]) != 'A' && toupper(rispostaUtente[0]) != 'B' && toupper(rispostaUtente[0]) != 'C') {
            printf("Risposta non valida. Inserisci A, B o C.\n");
        }
    } while (toupper(rispostaUtente[0]) != 'A' && toupper(rispostaUtente[0]) != 'B' && toupper(rispostaUtente[0]) != 'C');

    // Valuta la risposta
    if (toupper(rispostaUtente[0]) - 'A' == domande[i].rispostaCorretta) {
        printf("Corretto!\n");
        punteggio++;
    } else {
        printf("Sbagliato. La risposta corretta è %c\n", 'A' + domande[i].rispostaCorretta);
    }
}

// Stampa il punteggio finale
printf("\n%s, il tuo punteggio finale è: %d/5\n", nome, punteggio);

```

Gestione delle Risposte alle Domande:

All'interno del loop delle domande, ho migliorato la gestione dell'input utente utilizzando `fgets` per la lettura delle risposte. Inoltre, hai aggiunto un controllo per verificare che la risposta sia valida (A, B o C), evitando così input non desiderati.

Considerazioni Generiche:

L'importanza di gestire situazioni non standard è adottare un approccio proattivo per evitare potenziali errori logici. L'uso di `fgets` per la lettura delle stringhe è una buona pratica che contribuisce alla sicurezza del programma.