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.
Riprendete l'esempio di ieri ed identificate tutte le casistiche non contemplate. Provate a proporre un modello per gestirle modificando il codice sorgente del vostro programma. Aiutatevi pure con le risorse online, piccolo aiuto: cercate come gestire in maniera sicura l'input dell'utente (soprattutto quando parliamo di stringhe).

1. Gestione della Scelta Iniziale (A o B)
Attualmente, se l'utente inserisce una lettera diversa da 'A' o 'B', il programma ripete semplicemente il prompt. Questo approccio è corretto, ma possiamo migliorarlo per gestire anche input non validi come numeri o altri caratteri.
2. Gestione dell'Input del Nome
Se l'utente inserisce un nome che supera la dimensione dell'array name[50], il programma potrebbe comportarsi in modo imprevedibile, causando potenzialmente un buffer overflow. Per evitare ciò, utilizzeremo fgets() al posto di scanf() per limitare il numero di caratteri letti.
3. Gestione delle Risposte alle Domande
Se l'utente inserisce una risposta diversa da 'A', 'B', o 'C', il programma non gestisce correttamente l'errore. Aggiungeremo un controllo per verificare se l'input è valido e richiederemo nuovamente l'inserimento se non lo è.

Codice Modificato:

```c
#include <stdio.h>
#include <string.h>
#include <ctype.h>

#define MAX_NAME_LENGTH 50

// Variabile globale per mantenere il punteggio totale
int punteggioTotale = 0;

void startGame();
int askQuestion(const char *question, const char *options[], char correctAnswer);

int main() {
    char choice;
    printf("Benvenuto al gioco di domanda/risposta!\n");

    while (1) {
        printf("Scegli tra le seguenti opzioni:\n");
        printf("A) Iniziare una nuova partita\n");
        printf("B) Uscire dal gioco\n");
        printf("Inserisci la tua scelta (A/B): ");
        scanf(" %c", &choice);
        choice = toupper(choice);  // Convertiamo in maiuscolo per semplificare il confronto

        // Puliamo il buffer per evitare problemi con fgets successivo
        while (getchar() != '\n');

        if (choice == 'A') {
            startGame();
        } else if (choice == 'B') {
            printf("Il punteggio totale accumulato è: %d punti.\n", punteggioTotale);
            printf("Grazie per aver giocato! A presto!\n");
            break;
        } else {
            printf("Scelta non valida. Per favore, scegli tra A e B.\n");
            // Puliamo il buffer per evitare input indesiderati
            while (getchar() != '\n');
        }
    }

    return 0;
}

void startGame() {
    char name[MAX_NAME_LENGTH];
    printf("Inserisci il tuo nome: ");

    // Utilizziamo fgets per evitare il buffer overflow e rimuoviamo l'eventuale newline
    fgets(name, MAX_NAME_LENGTH, stdin);
    name[strcspn(name, "\n")] = '\0'; // Rimuove il carattere newline
```

```
/tmp/qgioTbNFWv.o
Benvenuto al gioco di domanda/risposta!
Scegli tra le seguenti opzioni:
A) Iniziare una nuova partita
B) Uscire dal gioco
Inserisci la tua scelta (A/B): D
Scelta non valida. Per favore, scegli tra A e B.
A
Scegli tra le seguenti opzioni:
A) Iniziare una nuova partita
B) Uscire dal gioco
Inserisci la tua scelta (A/B): A
Inserisci il tuo nome: Lucrezia
Ciao Lucrezia, iniziamo il gioco!
Qual è la capitale d'Italia?
A) Roma
B) Milano
C) Napoli
Inserisci la tua risposta (A/B/C): 3
Scelta non valida. Per favore, scegli tra A, B, o C.
Qual è la capitale d'Italia?
A) Roma
B) Milano
C) Napoli
Inserisci la tua risposta (A/B/C): A
Risposta corretta!
Qual è il pianeta più vicino al Sole?
A) Terra
B) Marte
C) Mercurio
Inserisci la tua risposta (A/B/C): B
Risposta sbagliata. La risposta corretta era C.
Qual è la lingua ufficiale del Brasile?
A) Spagnolo
B) Portoghese
C) Italiano
Inserisci la tua risposta (A/B/C): B
Risposta corretta!
Lucrezia, hai totalizzato un punteggio di 2 punti in questa partita.
Il punteggio totale accumulato è: 2 punti.
Scegli tra le seguenti opzioni:
A) Iniziare una nuova partita
B) Uscire dal gioco
Inserisci la tua scelta (A/B): B
Il punteggio totale accumulato è: 2 punti.
Grazie per aver giocato! A presto!


=== Code Execution Successful ===
```

```c

    printf("Ciao %s, iniziamo il gioco!\n", name);

    int punteggioPartita = 0;

    // Domanda 1
    const char *q1 = "Qual è la capitale d'Italia?";
    const char *q1_options[] = {"A) Roma", "B) Milano", "C) Napoli"};
    punteggioPartita += askQuestion(q1, q1_options, 'A');

    // Domanda 2
    const char *q2 = "Qual è il pianeta più vicino al Sole?";
    const char *q2_options[] = {"A) Terra", "B) Marte", "C) Mercurio"};
    punteggioPartita += askQuestion(q2, q2_options, 'C');

    // Domanda 3
    const char *q3 = "Qual è la lingua ufficiale del Brasile?";
    const char *q3_options[] = {"A) Spagnolo", "B) Portoghese", "C) Italiano"};
    punteggioPartita += askQuestion(q3, q3_options, 'B');

    punteggioTotale += punteggioPartita;

    printf("%s, hai totalizzato un punteggio di %d punti in questa partita.\n", name, punteggioPartita);
    printf("Il punteggio totale accumulato è: %d punti.\n", punteggioTotale);
}

int askQuestion(const char *question, const char *options[], char correctAnswer) {
    char answer;
    while (1) {
        printf("%s\n", question);
        printf("%s\n", options[0]);
        printf("%s\n", options[1]);
        printf("%s\n", options[2]);
        printf("Inserisci la tua risposta (A/B/C): ");
        scanf(" %c", &answer);
        answer = toupper(answer);  // Convertiamo in maiuscolo per semplificare il confronto

        if (answer == 'A' || answer == 'B' || answer == 'C') {
            if (answer == correctAnswer) {
                printf("Risposta corretta!\n");
                return 1;
            } else {
                printf("Risposta sbagliata. La risposta corretta era %c.\n", correctAnswer);
                return 0;
            }
        } else {
            printf("Scelta non valida. Per favore, scegli tra A, B, o C.\n");
            // Puliamo il buffer per evitare input indesiderati
            while (getchar() != '\n');
        }
    }
```

Modifiche e Gestione degli Errori

1. **Gestione della Scelta Iniziale**:
   - Convertiamo l'input dell'utente in maiuscolo (toupper(choice)) per semplificare i confronti.
   - Aggiungiamo un ciclo per pulire il buffer (while (getchar() != '\n');) quando l'utente inserisce un carattere non valido, così da evitare problemi con input successivi.
2. **Gestione dell'Input del Nome**:
   - fgets(name, MAX_NAME_LENGTH, stdin);: Utilizziamo fgets per leggere l'input dell'utente in modo sicuro, limitando il numero massimo di caratteri letti.
   - name[strcspn(name, "\n")] = '\0';: Rimuoviamo il newline \n che fgets aggiunge automaticamente alla fine della stringa, se c'è spazio nel buffer.
3. **Gestione delle Risposte alle Domande**:
   - Come per la scelta iniziale, convertiamo l'input dell'utente in maiuscolo (toupper(answer)) per facilitare i confronti.
   - Controlliamo se l'input è valido (ossia se è una delle opzioni 'A', 'B', 'C'). Se l'input non è valido, richiediamo nuovamente una risposta valida e puliamo il buffer.

Considerazioni Finali

- **Sicurezza dell'Input**: Utilizzando fgets e controlli sui caratteri, possiamo prevenire buffer overflow e garantire che gli input siano gestiti in modo sicuro e corretto.
- **Gestione degli Errori**: Abbiamo introdotto controlli per tutti i casi non contemplati, migliorando la robustezza del programma.
- **User Experience**: Fornire feedback immediato su input non validi e richiedere di riprovare migliora l'esperienza utente.