

```

#include <stdio.h>

void print_menu();
int gioca_partita();

int main(){
    char scelta={'\0'};
    print_menu();
    scanf ("%c", &scelta);
    if (scelta=='B'){
        printf("Grazie per aver giocato, alla prossima\n");
        return 0;
    }
    while (scelta=='A'){
        gioca_partita();
        print_menu();
        scanf ("%c", &scelta);
    }
    return 0;
}

void print_menu(){
    printf("Start menu:\n");
    printf("A >> Iniziare una nuova partita\nB >> Uscire dal gioco\n");
    printf("Inserisci la lettera corrispondente alla tua scelta");
}

int gioca_partita(){
    int punteggio=0;
    char nome[30]= {'\0'};
    char risposta1,risposta2;
    printf("Inserisci il tuo nome:\n");
    scanf("%s",&nome);

    printf("Domanda numero 1:\n");
    printf("Inserire qui la domanda\n");
    printf("A >> risposta 1\nB >> risposta 2\nC >> risposta 3\n");
    printf("Inserire la risposta:");
    scanf("%c", &risposta1);

    //gestiamo la casistica della risposta esatta, ipotizziamo sia la B, ed aggiungiamo 1 punto allo score totale
    if (risposta1=='B'){
        punteggio++;
    }

    printf("Domanda numero 2:\n");
    printf("Inserire qui la domanda\n");
    printf("A >> risposta 1\nB >> risposta 2\nC >> risposta 3\n");
    printf("Inserire la risposta:");
    scanf("%c", &risposta2);

    //gestiamo la casistica della risposta esatta per la seconda domanda, ipotizziamo sia la A, ed aggiungiamo 1 punto allo score totale
    if (risposta2=='A'){
        punteggio++;
    }

    printf("Partita conclusa, punteggio totalizzato da %s:%d\n", nome, punteggio);

    return 0;
}

```

Il programma, alla riga 3 richiama il **metodo print_menu**, che va a stampare le indicazioni che l'utente dovrà seguire se vuole iniziare una nuova partita (digitare A) o uscire dal gioco (digitare B). Alla riga 6, nel **metodo main**, abbiamo il comando if che dice: se l'utente digita B, il programma stamperà "Grazie per aver giocato, alla prossima". Poi abbiamo il ciclo while alla riga 15 che dice che, affinché l'utente digita A, ci rifacciamo al metodo

-gioca_partita: questo metodo setta la variabile punteggio uguale a 0, chiederà all'utente di scrivere il proprio nome e stamperà domanda numero 1, susseguito dalle varie risposte possibili indicate con A,B e C. Successivamente chiederà di inserire la nostra risposta. Alla riga 44 abbiamo un if, che dice che se la risposta alla domanda numero 1 è B, allora la variabile punteggio incrementa. Poi abbiamo la domanda numero 2 con anche qui le possibili risposte indicate A,B e C. Chiederà di

inserire la nostra risposta, e se (if) la risposta alla domanda numero 2 è A, allora la variabile punteggio incrementa. Infine stamperà che la partita è conclusa, facendo osservare il nome dell'utente e quanto abbia totalizzato.

Usciti dal metodo gioca_partita saremo alla riga 17 (sempre interna al while) dove abbiamo nuovamente il metodo print_menu, e possiamo così risSelectedinare se iniziare nuova partita (A) o uscire dal gioco (B).

Attualmente, il programma non gestisce correttamente l'input dell'utente se viene inserita una lettera diversa da A o B durante la scelta iniziale. Questo potrebbe portare a un comportamento imprevisto o a un loop infinito, poiché il programma non è progettato per gestire tali casi. Per migliorare il programma, potremmo considerare l'aggiunta di un'ulteriore logica per gestire gli input non validi, inserendo ad esempio un'istruzione **else** dopo l'if che controlla la scelta iniziale e se non è stato digitato né A e né B allora scriverà un messaggio di errore, chiedendo all'utente di inserire una scelta valida.

Se l'utente inserisce un nome che supera la dimensione dell'**array nome**, si verificherà un comportamento imprevisto noto come **buffer overflow**: è una vulnerabilità di sicurezza che si verifica quando un programma scrive più dati in un buffer di memoria di quanto esso possa gestire. (il **buffer** è un'area di memoria riservata per la conservazione temporanea di dati, e un **overflow** si verifica quando i dati superano la dimensione del buffer)

Questo può causare un comportamento indefinito nel programma e può portare a risultati imprevedibili. L'array nome è dichiarato come `char nome[20]`, il che significa che può contenere fino a 19 caratteri (considerando il terminatore di stringa `\0`, ovvero il punto in cui finisce la stringa). Se l'utente inserisce un nome più lungo di 19 caratteri, si verificherà un overflow del buffer.