

## 1. Cosa fa il programma?

Il programma sostanzialmente è un gioco in cui l'utente può iniziare una nuova partita (lettera A) oppure può uscire dal gioco (lettera B). Se l'utente sceglie di iniziare una nuova partita, gli verrà innanzitutto chiesto di inserire il suo nome, ma bisogna fare attenzione perché i caratteri a disposizione sono solo 20. Successivamente verranno poste due domande, ciascuna avente tre possibili risposte (A, B, C), e in caso di risposta esatta verrà aggiunto un punto al punteggio totale dell'utente. Una volta conclusa la partita, verrà stampata una frase contenente il nome dell'utente e il punteggio conseguito.

```
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[20] = {'\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
    if (risposta2 == 'A')
    {
        punteggio++;
    }

    printf ("Partita conclusa, punteggio totalizzato da %s:%d\n", nome, punteggio);
    return 0;
}
```

## 2. 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.

Se non si inserisce codice per gestire opzioni non valide, e l'utente inserisce una lettera diversa da A o B, il comportamento del programma sarà imprevedibile. Potrebbero verificarsi diverse situazioni:

- **Accesso a memoria non valida:** se si sta utilizzando l'input dell'utente per accedere a un'area di memoria basandosi sulla sua scelta, si potrebbe incorrere in errori di accesso a memoria non valida. Questo potrebbe causare il crash del programma o comportamenti imprevedibili.
- **Valori indeterminati:** se non si gestiscono le opzioni non valide, i valori delle variabili potrebbero rimanere indeterminati, portando a comportamenti imprevedibili.

- **Comportamento indefinito:** l'uso di valori non validi potrebbe portare a comportamenti indefiniti, il che significa che non si possono fare assunzioni affidabili sul comportamento del programma.

In linguaggio C, il termine "**comportamento imprevedibile**" si riferisce a situazioni in cui il risultato di un programma non può essere determinato in modo affidabile. Questo può accadere quando il programma viola le regole definite dallo standard C, portando a esiti che possono variare a seconda della scelta del compilatore o di sistemi operativi diversi.

3. 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).

Se l'utente inserisce un nome più lungo della dimensione dell'array "nome" precedentemente dichiarato, si verifica un overflow di array. Questo comportamento è "undefined" nel linguaggio C, il che significa che il compilatore e l'esecuzione del programma potrebbero comportarsi in modi imprevedibili. Quando si inserisce una stringa più lunga della dimensione dell'array, i caratteri in eccesso sovrascriveranno la memoria adiacente, portando a comportamenti indesiderati come la modifica di variabili o l'interruzione del normale flusso del programma. Questo può causare crash del programma o risultati inattesi. L'overflow di array è una fonte comune di errori e vulnerabilità in C, spesso utilizzata malevolmente dagli attacchi informatici. Per prevenire overflow di array, è importante gestire attentamente le dimensioni degli array e utilizzare funzioni sicure come "fgets" per leggere input utente.

```
kali@kali:~/Desktop
./Gioco
Start menu:
A >> Iniziare una nuova partita
B >> Uscire dal gioco
Inserisci la lettera corrispondente alla tua scelta:A
Inserisci il tuo nome:
aergjnalejirhtgbnaeilrjgbnsaerhjtgbakejrgbnaekrghj
Domanda numero 1:
Inserire qui la domanda
A >>> risposta 1
B >>> risposta 2
C >>> risposta 3
Inserire la risposta:Domanda numero 2:
Inserire qui la domanda
A >>> risposta 1
B >>> risposta 2
C >>> risposta 3
Inserire la risposta:A
Partita conclusa, punteggio totalizzato da aergjnalejirhtgbnaeilrjgbnsaerhjtgbakejr
gbnaekrghj:1785229926
zsh: segmentation fault ./Gioco
```

"Segmentation fault" è un errore molto comune in programmazione e indica che un programma ha tentato di accedere a una regione di memoria che non è stata assegnata a esso, o ha violato altre regole di accesso alla memoria.