

**Traccia: Riprendete il codice del programma che avete scritto e pensiamo all'ottimizzazione del codice alla gestione delle situazioni non previste 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.**

Una delle casistiche che non ho gestito nel codice precedente è: inserendo una lettera diversa da A e B stampare errore e ripresentare la scelta.

Ad esempio inserendo la lettera "D" all'inizio del programma questo era il risultato:

```
Benvenuto! Vuoi giocare?
A. Nuova partita
B. Esci dal gioco
D
Rispondi alle domande e cerca di indovinare più risposte possibili. Sei pronto?
Risposta non valida.
```

```
1 #include <stdio.h>
2
3 // Funzione che chiede all'utente se vuole giocare
4 char Iniziale()
5 {
6     printf ("Benvenuto! Vuoi giocare?\n");
7     printf ("A. Nuova partita\n");
8     printf ("B. Esci dal gioco\n");
9
10    char scelta;
11    scanf("%c", &scelta);
12    printf ("Rispondi alle domande e cerca di indovinare più risposte possibili. Sei pronto?\n");
13
14    if (scelta == 'A' || scelta == 'a')
15    {
16        printf ("Fantastico, hai scelto di iniziare una nuova partita.\n");
17    }
18    else if (scelta == 'B' || scelta == 'b')
19    {
20        printf ("Questo non è lo spirito giusto ;)\n");
21    }
22    else
23    {
24        printf ("Risposta non valida.\n");
25    }
26
27    return scelta;
28 }
29
```

Ho fatto anche delle modifiche per ottimizzare il codice come l'inserimento dei tre printf in una sola riga.

```
File Actions Edit View Help
#include <stdio.h>
#include <string.h> //inclusa libreria string
#include <stdbool.h> //inclusa libreria booleana

void pulisciBuffer() {
    int c;
    while ((c = getchar()) != '\n' && c != EOF); //Per evitare vada in loop
}

// Funzione che chiede all'utente se vuole giocare
char Iniziale()
{
    printf ("Benvenuto! Vuoi giocare?\nA. Nuova partita\nB. Esci dal gioco\n");

    char scelta;

    scanf("%c", &scelta);
    //printf ("Rispondi alle domande e cerca di indovinare più risposte possibili. Sei pronto?\n");

    if (scelta == 'A' || scelta == 'a'){
        printf ("Fantastico, hai scelto di iniziare una nuova partita.\n");
    } else if (scelta == 'B' || scelta == 'b'){
        printf ("Questo non è lo spirito giusto\n");
    } else {
        printf ("Risposta non valida.\n");
    }

    return scelta;
}
```

Il nuovo codice risponde così, ho corretto il fatto che venisse presentata la frase di inizio del gioco anche in caso di errore (come l'inserimento della lettera D):

```
Benvenuto! Vuoi giocare?
A. Nuova partita
B. Esci dal gioco
D
Risposta non valida.
Non andare via. Gioca ancora! Pronto?
S- Per continuare
N- Per chiudere
La tua scelta è: 
```

**-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)**

Nel codice precedente non avevo inserito un limite di caratteri per l'inserimento del nome quindi succedeva questo:

```
Come ti chiami?: bchifewbvcwksjvncodlwrbvnujsdohdjncueknsikf
CIAO bchifewbvcwksjvncodlwrbvnujsdohdjncueknsikf
Iniziamo!
```

```
29
30 // Funzione per chiedere il nome del giocatore
31 void NuovaPartita(char *NomeGiocatore)
32 {
33     printf("Come ti chiami?: ");
34     scanf("%s", NomeGiocatore);
35     printf("CIAO %s\n", NomeGiocatore);
36 }
37
```

```
// Funzione per chiedere il nome del giocatore
void NuovaPartita(char *NomeGiocatore){
    bool errore;

    do {
        printf("Inserisci il tuo nome: ");
        scanf("%9s", NomeGiocatore); // Limite impostato a 10 per evitare buffer overflow

        if (strlen(NomeGiocatore) > 8) { // Limite massimo tramite libreria stringa
            printf("Errore: nome troppo lungo. Riprova\n");
            pulisciBuffer(); // Pulizia buffer in caso di loop del programma
            errore = true; // In caso di errore (nome troppo lungo) ricarica nuovamente la domanda
        } else {
            printf("Quindi ti chiami: %s\n\n", NomeGiocatore); // (nome di lunghezza sotto i 10 caratteri)
            errore = false; // In questo caso la stringa booleana non richiama il checkpoint/ciclo essendo il nome entro i 10 caratteri
        }
    } while (errore);
}
```

Ho aggiunto un limite di caratteri (8) per l'inserimento del nome e la possibilità di ricevere un messaggio di errore per poi inserire nuovamente il nome corretto:

```
Benvenuto! Vuoi giocare?
A. Nuova partita
B. Esci dal gioco
A
Fantastico, hai scelto di iniziare una nuova partita.
Inserisci il tuo nome: hdnegjfmtdthcnhgkjedudejhjfsjchdjksdx
Errore: nome troppo lungo. Riprova
Inserisci il tuo nome: 
```

## -Cosa succede se l'utente inserisce la lettera D per la risposta alle domande durante una partita? O un carattere numerico?

Nel mio codice precedente avevo pensato a questa casistica e avevo inserito “printf (“Ops, puoi rispondere solo con A-B-C:”);” tutte le volte che la scelta non fosse stata A,B,C o a,b,c. Ma nel caso del carattere numerico la frase viene stampata in loop.

```
Seconda domanda:  
Dove si trova la sede della Apple?  
A. SACRAMENTO  
B. SANTA MONICA  
C. CUPERTINO  
SCELTA: D  
Ops, rispondi con A,B o C:|
```

```
Cosa significa la sigla OSINT?  
A. OPEN SOURCE INTELLIGENCE  
B. OPEN SOFTWARE INTELLIGENCE  
C. OPEN SOFTWARE INTEL NET  
SCELTA: 3657  
Ops, rispondi con A,B o C:Ops, rispondi con A,B o C:Ops, rispondi con A,B o C:Ops,  
rispondi con A,B o C:
```

Avendo già considerato questa casistica in precedenza non ho fatto grandi modifiche al codice, ora succede questo:

```
Dove si trova la sede della Apple?  
A. SACRAMENTO  
B. SANTA MONICA  
C. CUPERTINO  
SCELTA: D  
Ops, rispondi con A,B o C:█
```

```
Terza domanda:  
Cosa significa la sigla OSINT?  
A. OPEN SOURCE INTELLIGENCE  
B. OPEN SOFTWARE INTELLIGENCE  
C. OPEN SOFTWARE INTEL NET  
SCELTA: 4  
Ops, rispondi con A,B o C:█
```

In tutte le domande ho apportato questa piccola modifica per maggiore correttezza:

```
while (Scelta != 'A' && Scelta != 'B' && Scelta != 'C' && Scelta != 'a' && Scelta != 'b' && Scelta != 'c')
```

```
while (Scelta ≠ 'A' && Scelta ≠ 'B' && Scelta ≠ 'C' && Scelta ≠ 'a' && Scelta ≠ 'b' && Scelta ≠ 'c')
```

Ho infine modificato la parte finale del codice:

```
// Chiedi se vuole giocare di nuovo
printf ("\n Non andare via. Gioca ancora! Pronto?");
scanf( "%c", &scelta);
```

```
    // restituisce il punteggio ottenuto
    printf ("Il tuo punteggio finale è: %d\n", Punteggio);
}

int main ()
{
    char scelta;
    char NomeGiocatore [10];

    do {
        // Chiamata alla funzione per chiedere se si vuole giocare
        scelta = Iniziale();

        if (scelta == 'A' || scelta == 'a')
        {
            // Se l'utente sceglie di giocare, chiedi il nome del giocatore
            NuovaPartita(NomeGiocatore);
            // Esegui il quiz
            setDiDomande();
        }
        // Chiedi se vuole giocare di nuovo
        printf ("Non andare via. Gioca ancora! Pronto?\nS- Per continuare\nN- Per chiudere\nLa tua scelta è: ");
        scanf( " %c", &scelta);

    } while (scelta == 'S' || scelta == 's');

    return 0;
}
```