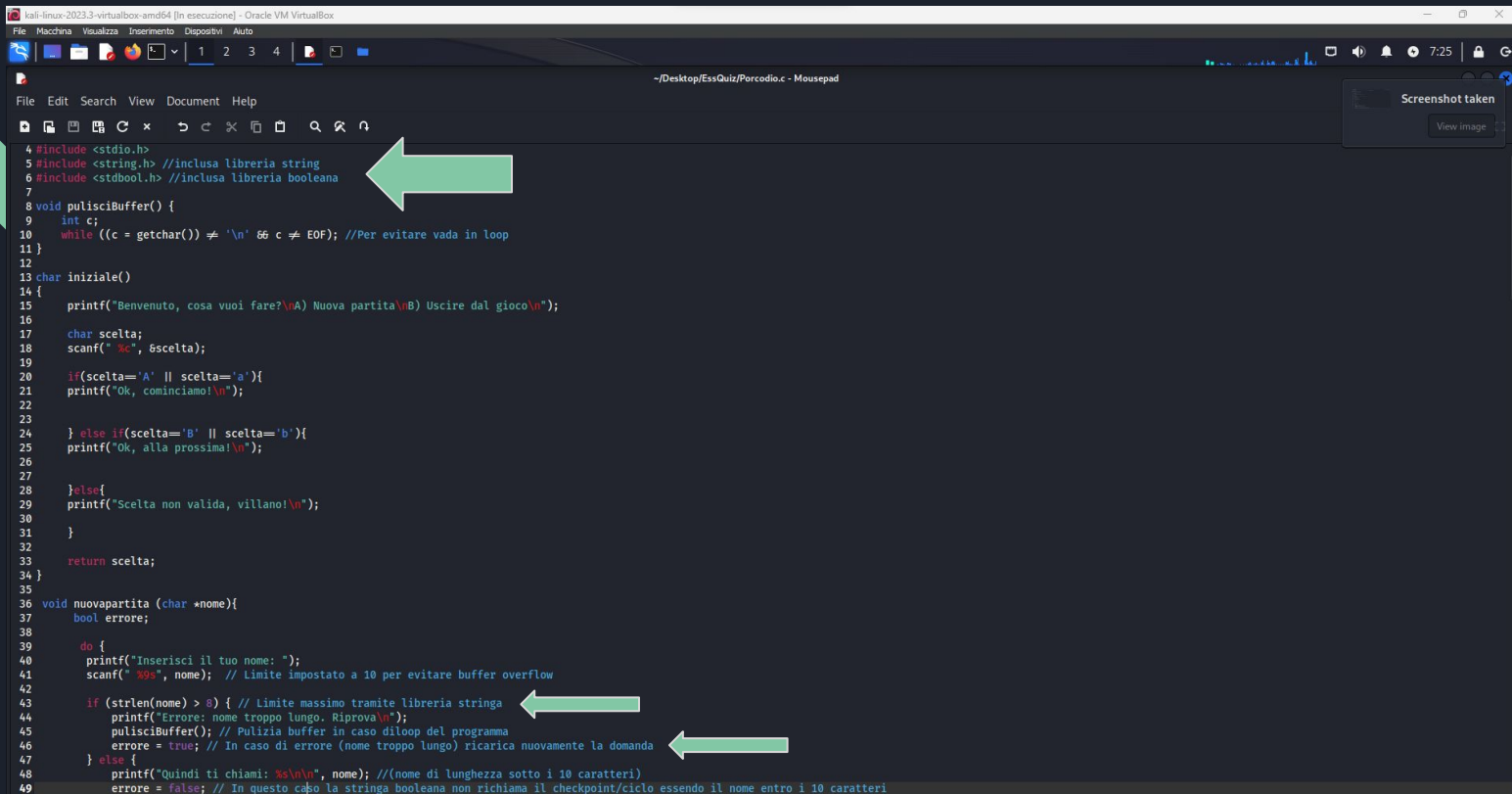


# Esercizio 2 S2L5

## Miglioramento

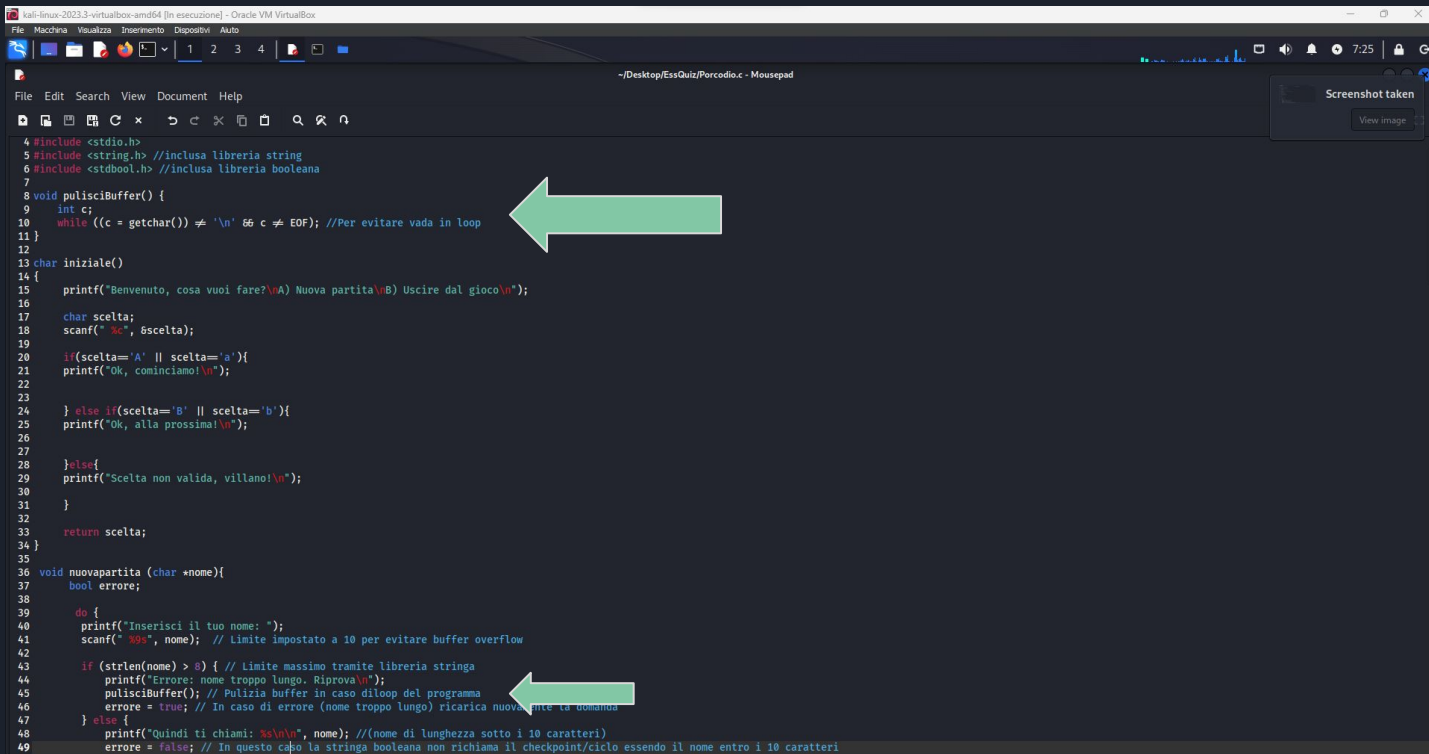

## Codice



The screenshot shows a code editor window titled "kali-linux-2023.3-virtualbox-amd64 [in esecuzione] - Oracle VM VirtualBox". The editor displays C code for a game. Two green arrows point to specific lines: one points to line 4, which includes `<stdio.h>`, and the other points to line 43, which includes `<string.h>`. The code includes `<string.h>` and `<stdbool.h>` for string and boolean operations. The `pulisciBuffer()` function is defined to clear the input buffer. The `iniziale()` function handles the initial game state. The `nuovapartita()` function handles the start of a new game, including a loop to ensure the name is within 10 characters. The code uses `strlen` to check the length of the name and `bool` to manage the game state.

```
4 #include <stdio.h>
5 #include <string.h> //inclusa libreria string
6 #include <stdbool.h> //inclusa libreria booleana
7
8 void pulisciBuffer() {
9     int c;
10    while ((c = getchar()) != '\n' && c != EOF); //Per evitare vada in loop
11 }
12
13 char iniziale()
14 {
15     printf("Benvenuto, cosa vuoi fare?\nA) Nuova partita\nB) Uscire dal gioco\n");
16
17     char scelta;
18     scanf(" %c", &scelta);
19
20     if(scelta=='A' || scelta=='a'){
21         printf("Ok, cominciamo!\n");
22     }
23
24     } else if(scelta=='B' || scelta=='b'){
25         printf("Ok, alla prossima!\n");
26     }
27
28     }else{
29         printf("Scelta non valida, villano!\n");
30     }
31 }
32
33 return scelta;
34 }
35
36 void nuovapartita (char *nome){
37     bool errore;
38
39     do {
40         printf("Inserisci il tuo nome: ");
41         scanf(" %9s", nome); // Limite impostato a 10 per evitare buffer overflow
42
43         if (strlen(nome) > 10) { // Limite massimo tramite libreria stringa
44             printf("Errore: nome troppo lungo. Riprova!\n");
45             pulisciBuffer(); // Pulizia buffer in caso di loop del programma
46             errore = true; // In caso di errore (nome troppo lungo) ricarica nuovamente la domanda
47         } else {
48             printf("Quindi ti chiami: %s\n", nome); //(nome di lunghezza sotto i 10 caratteri)
49             errore = false; // In questo caso la stringa booleana non richiama il checkpoint/ciclo essendo il nome entro i 10 caratteri
```


Qui come indicato nei commenti in confronto al codice precedente abbiamo incluso nuove librerie, la libreria `<string.h>` per poter utilizzare il comando `strlen` per impostare un numero massimo di caratteri utilizzabili, la seconda, la libreria `bool` per aggiungere una sorta di checkpoint che in caso di errore di inserimento (ad esempio un nome troppo lungo) ricarichi la domanda.



```
1 #include <stdio.h>
2 #include <string.h> //includa libreria string
3 #include <stdbool.h> //includa libreria booleana
4
5 void puliscibuffer() {
6     int c;
7     while ((c = getchar()) != '\n' && c != EOF); //Per evitare vada in loop
8 }
9
10 char iniziale()
11 {
12     printf("Benvenuto, cosa vuoi fare?\nA) Nuova partita\nB) Uscire dal gioco\n");
13
14     char scelta;
15     scanf(" %c", &scelta);
16
17     if(scelta=='A' || scelta=='a'){
18         printf("Ok, cominciamo!\n");
19     } else if(scelta=='B' || scelta=='b'){
20         printf("Ok, alla prossima!\n");
21     } else{
22         printf("Scelta non valida, villano!\n");
23     }
24
25     return scelta;
26 }
27
28 void nuovapartita (char *nome){
29     bool errore;
30
31     do {
32         printf("Inserisci il tuo nome: ");
33         scanf(" %s", nome); // Limite impostato a 10 per evitare buffer overflow
34
35         if (strlen(nome) > 10) { // Limite massimo tramite libreria stringa
36             printf("Errore: nome troppo lungo. Riprova!\n");
37             puliscibuffer(); // Pulizia buffer in caso di loop del programma
38             errore = true; // In caso di errore (nome troppo lungo) ricarica nuovamente la domanda
39         } else {
40             printf("Quindi ti chiami: %s\n", nome); //(nome di lunghezza sotto i 10 caratteri)
41             errore = false; // In questo caso la stringa booleana non richiama il checkpoint/ciclo essendo il nome entro i 10 caratteri
42         }
43     } while (errore);
44 }
```

Abbiamo poi inserito la funzione << Void puliscibuffer>> per evitare che il programma vada in loop.

Funzione che abbiamo poi richiamato nel do while della funzione <<Void Nuova Partita>>



```
kali-linux-2023.3-virtualbox-guest64 [in esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Auto
1 2 3 4
~/Desktop/EsQuiz/Porcodio.c - Mousepad

File Edit Search View Document Help
printf("Quindi ti chiami: %s\n", nome); //(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);
}
52
53
54 void domande(){
55     char domanda1[]="Quante stelle ci sono nel cielo alle 10 del mattino? ....\n";
56     char domanda2[]="Di che colore era il cavallo bianco di Napoleone? ....\n";
57     char domanda3[]="Quanto tempo ti serve per riempire di acqua uno scola-pasta? ....\n";
58     char domanda4[]="Se tu avessi l'alopecia quanti capelli avresti in testa? ....\n";
59     char domanda5[]="Come si chiama il dito fra medio e mignolo? ... \n";
60
61
62     int score = 0;
63     char scelta;
64     printf("Cominciamo la partita!\n");
65
66     //Inizio Domanda 1
67     printf("Prima domanda.....\n");
68     printf(domanda1);
69     printf("A) 10 — B) 312,3 — C) ma è giorno... \n");
70     printf("Ora scegli\n");
71     scanf("%c", &scelta);
72
73     while(scelta != 'A' && scelta != 'B' && scelta != 'C' && scelta != 'a' && scelta != 'b' && scelta != 'c'){ //Aggiunta funzione per far sì che non ci sia distizione
74         printf("Scelta invalida\n");
75         //pulisciBuffer();
76         scanf("%c", &scelta);
77     }
78
79     if(scelta == 'A' || scelta == 'B' || scelta == 'a' || scelta == 'b'){
80         printf("Risposta sbagliata, perdi 1 punto!\n");
81         score--;
82     }
83     else {
84         printf("Risposta corretta + 1 punto!\n");
85         score++;
86     }
87
88     //Inizio Domanda 2
89     printf("Seconda domanda.....\n");
90     printf(domanda2);
91     printf("A) Viola — B) Bianco — C) Arcobaleno \n");
92     printf("Ora scegli\n");
93     scanf("%c", &scelta);
```



Abbiamo creato un array contenesse tutte le domande per risparmiare righe di codice senza dover fare un printf per ogni domanda, come possiamo notare il comando pulisci buffer è presente dopo ogni ciclo.

```

175 }
176 }
177
178 int main() {
179     char scelta;
180     char nome[10];
181
182     do{
183         scelta=iniziale();
184
185         if (scelta=='A' || scelta=='a'){
186             nuovapartita(nome);
187             domande();
188         }
189     } while(scelta=='A' || scelta=='a');
190     printf("Vuoi giocare di nuovo s/n \n");
191     scanf(" %c", &scelta);
192
193
194
195 }
196
197 return 0;
198 }
199
200

```

Questa è l'ultima funzione che permette di cominciare una nuova partita che permetterà di inserire un nuovo nome e ricominciare il ciclo da capo.