

## Consegna:

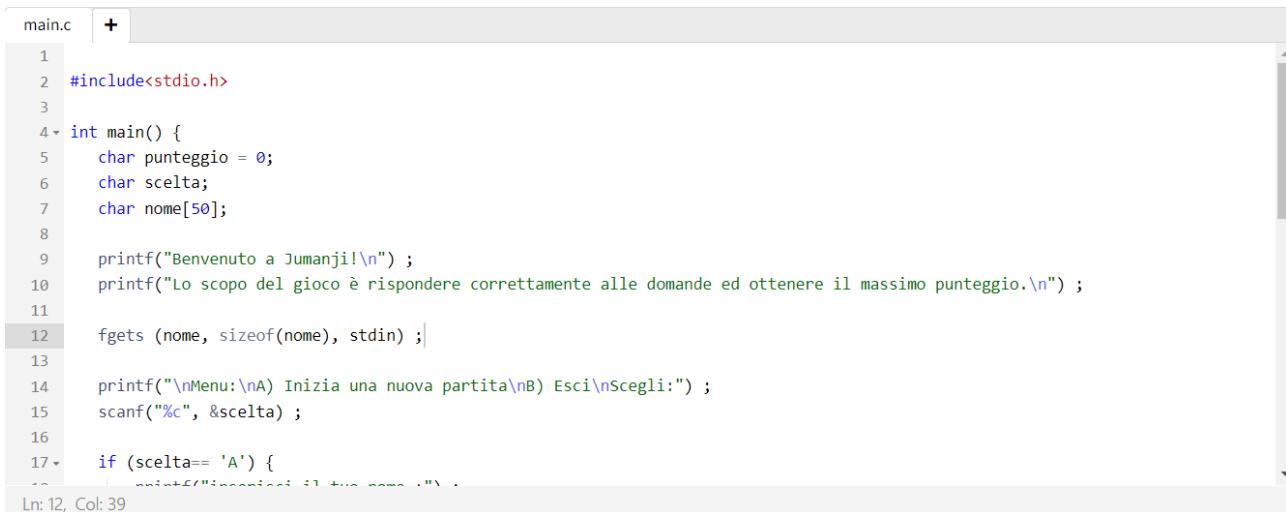


**Esercizio**  
Ottimizzazione del codice

**Traccia:**  
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).

## Soluzione:



```
main.c +  
1  
2 #include<stdio.h>  
3  
4 int main() {  
5     char punteggio = 0;  
6     char scelta;  
7     char nome[50];  
8  
9     printf("Benvenuto a Jumanji!\n");  
10    printf("Lo scopo del gioco è rispondere correttamente alle domande ed ottenere il massimo punteggio.\n");  
11  
12    fgets (nome, sizeof(nome), stdin);  
13  
14    printf("\nMenu:\nA) Inizia una nuova partita\nB) Esci\nScegli:");  
15    scanf("%c", &scelta);  
16  
17 if (scelta== 'A') {  
18     printf("Inizierai la tua partita.\n");  
Ln: 12, Col: 39
```

### Parametri di fgets:

- **nome:** Questo è il nome dell'array di caratteri in cui verrà memorizzata la stringa letta da **fgets**. Bisogna assicurarsi che l'array abbia sufficiente spazio per contenere la stringa che prevediamo di leggere.
- **sizeof(nome):** Questa è la dimensione in byte dell'array **nome**. **sizeof** è un operatore che restituisce il numero totale di byte occupati dall'array. In questo modo, **fgets** sa quanto spazio è disponibile nell'array.
- **stdin:** Questo è il file stream di input standard, associato di solito alla tastiera. Indica da dove leggere l'input.

Quindi, **fgets(nome, sizeof(nome), stdin);** legge una linea da tastiera e la memorizza nell'array **nome**, con la massima lunghezza di **sizeof(nome) - 1** caratteri (l'ultimo carattere è riservato per il terminatore nullo '\0', che indica la fine della stringa).