

# Domino Lineare

Il Domino Lineare si gioca usando di 21 tessere, ciascuna con una coppia di numeri da 1 a 6. Se rappresentiamo una tessera come una coppia di numeri tra parentesi quadre, allora tutte le tessere del gioco sono le seguenti.

```
[1|1] [1|2] [1|3] [1|4] [1|5] [1|6]
[2|2] [2|3] [2|4] [2|5] [2|6]
[3|3] [3|4] [3|5] [3|6]
[4|4] [4|5] [4|6]
[5|5] [5|6]
[6|6]
```

Al giocatore vengono assegnate  tessere di questo tipo, anche con *ripetizioni*, che deve disporre su un piano di gioco orizzontale.

Il suo obiettivo è disporle sul piano di gioco (inizialmente vuoto) totalizzando il massimo punteggio e seguendo la seguente regola: **i numeri dei lati adiacenti tra due tessere devono essere identici.**

Ad esempio, il giocatore potrebbe ricevere le seguenti  tessere:

```
[6|6] [6|6] [6|6] [3|6] [1|6] [2|6] [1|4] [3|4] [4|4] [4|5]
```

Supponiamo che il giocatore scelga come prima la seguente tessera:

```
[6|6]
```

**Le prossime tessere possono essere poste sia sul lato destro che sul lato sinistro.** Ad esempio, il giocatore può aggiungere la tessera  a sinistra e un'altra tessera  a destra come sotto.

```
[2|6][6|6][6|6]
```

Il giocatore può anche "ruotare" le tessere in suo possesso. Ad esempio aggiungendo la tessera  a destra opportunatamente ruotata in , come mostrato sotto.

```
[2|6][6|6][6|6][6|1]
```

**Il gioco termina quando il giocatore non può più posizionare tessere.**

**Il punteggio del giocatore è dato dalla somma delle cifre nelle tessere poste sul piano di gioco.** Se il gioco terminasse qui, il punteggio sarebbe quindi di  $2 + 6 + 6 + 6 + 6 + 6 + 1 = 39$  punti.

## Modalità interattiva

---

Viene chiesto all'utente quale sia la mossa successiva. Il piano di gioco ed il punteggio corrente vengono aggiornati di conseguenza. Il gioco termina automaticamente se il giocatore non ha più mosse valide disponibili.

## Modalità AI

---

Si richiede di implementare una funzione che calcoli una buona strategia di gioco.

### Strategia 1: First Match

Provo tutte le tessere in mio possesso e posiziono la prima che abbia un match.

### Altre Strategie

A voi piena libertà. Ad esempio preferire le tessere che danno un punteggio più alto, o con cifre che appaiono frequentemente in altre tessere in possesso del giocatore. Un'opzione è una soluzione ricorsiva che implementi la ricerca esaustiva della migliore configurazione possibile.

## Partite Speciali

---

Potete pensare a partite che iniziano da un insieme specifico di pezzi per il giocatore (e non casuale). Durante il corso verranno pubblicati alcuni di questi insiemi e vi verrà chiesto di trovare la soluzione migliore.

## Schema di implementazione

---

Il programma che implementerete dovrà chiedere al giocatore quale mossa giocare, e dovrà visualizzare il campo di gioco aggiornato (usando la funzione `printf`). Dovrà verificare che la mossa scelta sia legale (ci sia un match tra le cifre), e dovrà aggiornare il punteggio. Questi passi si ripetono fino alla fine della partita.

Nella modalità AI, il programma dovrà calcolare le mosse migliori, e visualizzarle a schermo assieme al piano di gioco finale.

## Consegna

**Quando?** Il progetto deve essere consegnato 4 giorni prima la data di ciascun appello. E' sempre obbligatorio iscriversi all'appello di "Esercizi" del prof. Spanò per sostenere la discussione del progetto.

**Dove?** Il docente abiliterà la consegna tramite Moodle secondo le scadenze previste.

**Cosa?** Dovrete consegnare un unico file `.zip` contenente:

1. Una relazione scritta di al massimo 3 pagine che descriva la struttura del vostro progetto, l'organizzazione del lavoro tra i componenti del gruppo, le principali difficoltà incontrate. (Relazioni più lunghe verranno penalizzate.)
2. Il codice sorgente del progetto in linguaggio C99.
3. Documentazione delle funzioni, dei tipi e dei file generata con [Doxygen](#).

## Librerie esterne

---

E' possibile implementare l'intero gioco utilizzando semplicemente la standard library di C, usando le funzioni di lettura dei tasti come `scanf`, `getc` o `getchar` e le funzioni di stampa come `printf`. Coloro che vogliono cimentarsi nell'uso di librerie esterne per dare una veste grafica ed una interazione col giocatore più accattivanti e fluide possono farlo, tuttavia sarà propria responsabilità occuparsi degli aspetti di portabilità e compatibilità. Il programma deve compilare e funzionare correttamente sulle 3 piattaforme (Windows, Linux, Mac); in caso contrario, è necessario documentare e motivare opportunamente la scelta nella relazione che accompagnerà il progetto ed in sede d'esame.

## Valutazione

Per i progetti di *gruppo*:

- progetto **sufficiente** se permette ad un utente di giocare inserendo le proprie mosse tramite input da tastiera (modalità interattiva); l'implementazione delle regole del gioco deve essere corretta.
- progetto **buono** se implementa la modalità AI utilizzando un algoritmo semplice; avete piena libertà di definire una strategia.
- progetto **ottimo** se implementa una strategia particolarmente interessante, come ad esempio un **algoritmo ricorsivo**, per la modalità AI.

**Tutti i membri del gruppo devono conoscere ogni riga del codice.**

Nel caso di progetti *individuali*, ad esempio per studenti lavoratori:

- progetto **sufficiente** se permette ad un utenti di giocare inserendo le proprie mosse tramite input da tastiera; l'implementazione delle regole del gioco può essere "parziale".
- progetto **buono** se permette ad un utente di giocare inserendo le proprie mosse tramite input da tastiera; l'implementazione delle regole deve essere completa e corretta.
- progetto **ottimo** se implementa una strategia in modalità AI; avete piena libertà di definire una

strategia.

**Importante:** Il progetto è di gruppo, ma **la valutazione è individuale**. Questo significa che i componenti di un gruppo potrebbero ricevere un voto diverso.

Inoltre, è possibile per alcuni dei componenti presentare individualmente delle **migliorie** al progetto.

Possibili migliorie potrebbero essere: grafica migliore, strategia di gioco più raffinata, menu di interazione dell'utente più usabile, etc.