

# Esercitazione 10

20 gennaio 2015

Termine per la consegna dei lavori: **martedì 27 gennaio ore 23.59.**

## Istruzioni

I lavori dovranno essere salvati in una cartella che deve contenere tutto e solo ciò che volete venga consegnato e valutato, d'ora in poi sarà **obbligatoriamente** un file **eseguibile** con estensione `.py` per ognuno degli esercizi. Controllate che l'esecuzione del comando:

```
python <nome_file>.py
```

per ognuno degli esercizi produca l'output desiderato.

Per consegnare gli elaborati dovete raggiungere la cartella contenente i file da inviare in modalità terminale (`cd path_della_cartella`) e quindi eseguire il comando:

```
consegna consegna10
```

verrà visualizzata la lista di tutto ciò che è stato inviato.

Consegne successive (entro il termine per la consegna) sovrascriveranno le precedenti, verrà valutata solo l'ultima consegna sottomessa.

È obbligatorio che all'interno di ogni file sia riportato il vostro nome, cognome e numero di matricola (riportati all'interno di una riga commento all'inizio del file, es: `#Mario Rossi 1234567`).

## ATTENZIONE!

L'unico modulo importabile ammesso in questa esercitazione è il modulo `random`. Esercizi risolti utilizzando altri moduli importati riceveranno il punteggio minimo. Python contiene delle **built-in functions**, che potete utilizzare e la cui lista si può trovare a questo indirizzo: <https://docs.python.org/2/library/functions.html>.

## NOTA BENE!!!

Essendo l'ultima esercitazione vi viene data la **possibilità di scegliere una tra le due esercitazioni seguenti: Esercitazione A oppure Esercitazione B.**

### Esercitazione A

#### Esercizio A1: Il problema delle 8 regine

Il problema delle otto regine è un problema che consiste nel trovare il modo di posizionare otto regine (pezzo degli scacchi) su una scacchiera 8x8 tali che nessuna di esse possa catturarne un'altra, usando i movimenti standard della regina. Per maggiori informazioni guardare la pagina wikipedia [http://it.wikipedia.org/wiki/Rompicapo\\_delle\\_otto\\_regine](http://it.wikipedia.org/wiki/Rompicapo_delle_otto_regine).

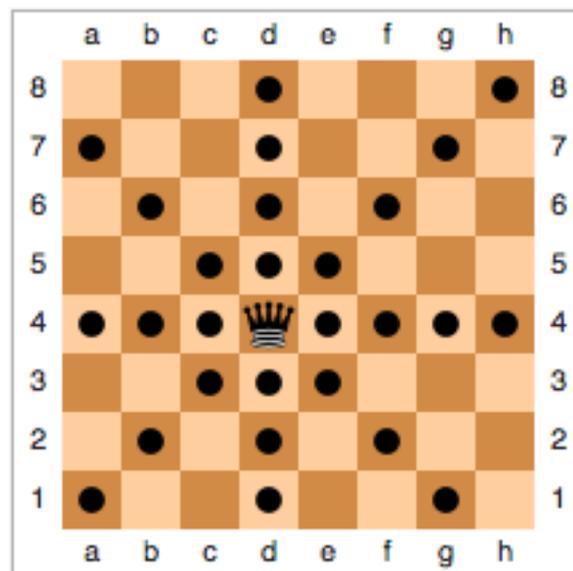


Figura 1: Movimenti standard di una regina.

Realizzare una (o più) funzione che risolve il problema delle 8 regine in modo **ricorsivo**, stampando a video la soluzione trovata.

L'algoritmo è basato sul backtracking: partendo dalla scacchiera vuota, si posiziona una prima regina nella prima colonna e poi si passa alla successiva. Fintanto che nella  $i$ -esima colonna (con le  $i - 1$  precedenti aventi le regine posizionate) non si trova una casella che rispetti le regole si sposta la regina alla riga successiva. Se nessuna riga della colonna in esame è valida, tornare alla colonna, e quindi regina, precedente (passo di backtracking) ripetendo il meccanismo.

## Esercizio A2: Quicksort

Il quicksort è un algoritmo di ordinamento che si basa sul paradigma divide et impera (come il merge sort). Considerando un array di valori interi l'algoritmo opera come segue:

1. seleziona un elemento casuale, detto perno;
2. posta tutti gli elementi inferiori del perno alla sua sinistra mentre quelli maggiori alla sua destra all'interno dell'array. In questo modo il perno si troverà nella sua posizione finale;
3. ripete la procedura (dal punto 1) nelle due sotto parti dell'array costituite dai valori alla destra e alla sinistra del perno.

Per maggiori dettagli sull'algoritmo vedere la pagina wikipedia <http://it.wikipedia.org/wiki/Quicksort>. Si chiede di implementare una o più funzioni che realizzino l'algoritmo quicksort in modo **ricorsivo**.

## Esercitazione B

### Esercizio B1: Scacchi - Finale Torre e Re contro Re

Realizzare uno script python per giocare un finale di scacchi Torre e Re contro Re con il computer. Per una visione generale delle regole del gioco degli scacchi vedere la pagina wikipedia <http://it.wikipedia.org/wiki/Scacchi>. Ai fini di questo esercizio però è sufficiente conoscere le seguenti regole:

1. **gioco a turni**: gli scacchi, come la dama, è un gioco a turni e durante ogni turno il giocatore a cui spetta muovere può eseguire una sola mossa;
2. **concetto di Scacco**: quando, una volta mosso, il Re avversario è direttamente attaccato da un nostro pezzo allora l'avversario è sotto Scacco e deve per forza di cose uscire da tale situazione (o muovendo il Re o tramite l'uso di un altro pezzo). Se nessuna mossa legale può evitare la situazione di Scacco allora lo Scacco è matto (punto 3) (vedi figura 2);
3. **scopo del gioco**: negli scacchi per vincere bisogna catturare il Re avversario. Tale cattura deve avvenire in maniera forzata e non può essere agevolata dallo stesso avversario. Quando, una volta mosso, il Re avversario è sotto scacco e qualsiasi mossa legale non può evitare tale situazione, allora la partita è vinta per Scacco Matto;
4. **la cattura**: una cattura può essere effettuata se un pezzo avversario è nel raggio d'azione di un proprio pezzo. Il pezzo che cattura si posizionerà nella casella occupata dal pezzo catturato;
5. **movimento del Re**: il Re può muovere su tutte le case adiacenti (a distanza 1) a quella in cui è situato (anche diagonalmente) (vedi figura 3);

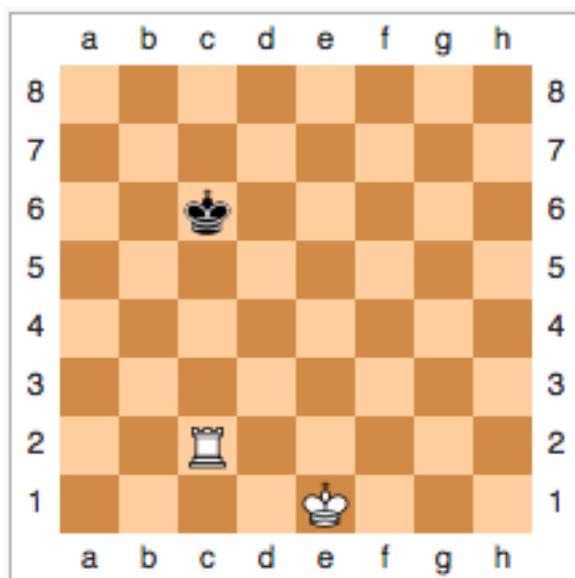


Figura 2: Esempio di scacco: la mossa è al nero.

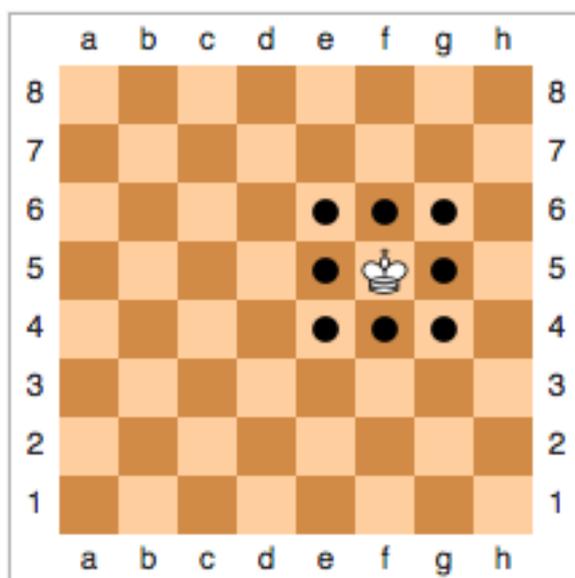


Figura 3: Esempio di movimento del Re.

6. **movimento della Torre:** la Torre può muovere in qualsiasi casa posta nella stessa colonna o nella stessa traversa (o riga) in cui si trova. N.B. la Torre non può saltare i pezzi quindi il suo raggio è limitato sia dai propri che dai pezzi avversari con la differenza che un pezzo avversario può essere catturato (vedi figura 4);
7. **il pareggio:** una partita a scacchi può terminare in parità se:
  - **insufficienza di materiale:** non ci sono pezzi sufficienti per dare scacco matto (per nessuno dei due contendenti) e.g., Re contro Re;

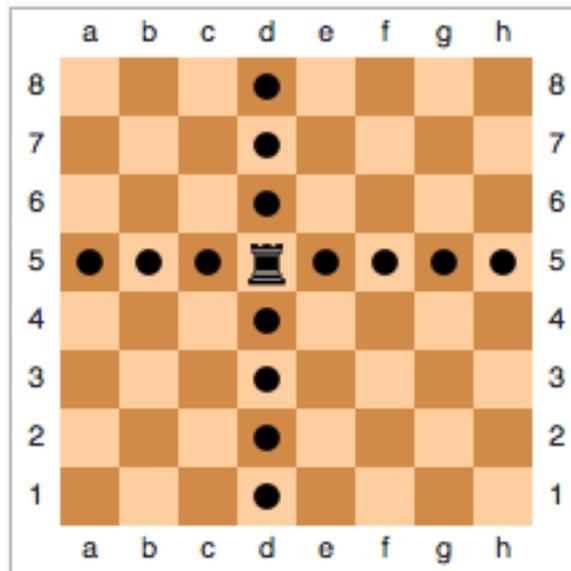


Figura 4: Esempio di movimento della Torre.

- **stallo**: se tutte le mosse legali possibile per il giocatore che ha il turno portano ad una situazione di scacco per se stesso, mentre la posizione attuale non è una posizione di scacco, allora è stallo;
- **regola delle 50 mosse**: se nessuno dei contendenti vince la partita dopo 50 mosse (100 semimosse) consecutive in cui non vengono catturati pezzi e non vengono mossi pedoni allora questa viene definita patta;
- **ripetizione**: se la stessa identica posizione si ripete sulla scacchiera 3 volte, anche non consecutivamente, durante la partita allora questa è definita patta.

Quello che è richiesto in questa esercitazione è di creare il gioco degli scacchi limitato al solo finale Re e Torre contro Re. Per maggiori dettagli vedere [http://it.wikipedia.org/wiki/Finali\\_elementari#Re\\_e\\_torre\\_contro\\_re](http://it.wikipedia.org/wiki/Finali_elementari#Re_e_torre_contro_re). Sostanzialmente nella scacchiera compariranno per il bianco il Re e una Torre, mentre per il nero il solo Re (vedi figura 5).

Una nuova partita posizionerà questi pezzi in modo casuale sulla scacchiera in modo che tutte le regole siano rispettate e il turno spetterà al giocatore umano con i pezzi bianchi. Il gioco poi si svolgerà a turni con le regole sopra descritte e lo scopo per chi gioca con il bianco è di dare scacco matto evitando di cadere in una delle casistiche di pareggio descritte.

Per l'esercizio viene data completa libertà sulla gestione della scacchiera e dei pezzi. Viene però richiesta la creazione di una classe che gestisca il gioco, la quale **deve** contenere:

- una funzione di stampa per poter visualizzare a video la posizione attuale;
- una funzione per ricevere in input le mosse dell'utente: si consiglia di usare la notazione "battaglia navale like", ad esempio, "e2e4" per dire che si vuole muovere il pezzo che si trova nella casa "e2" alla casa "e4";

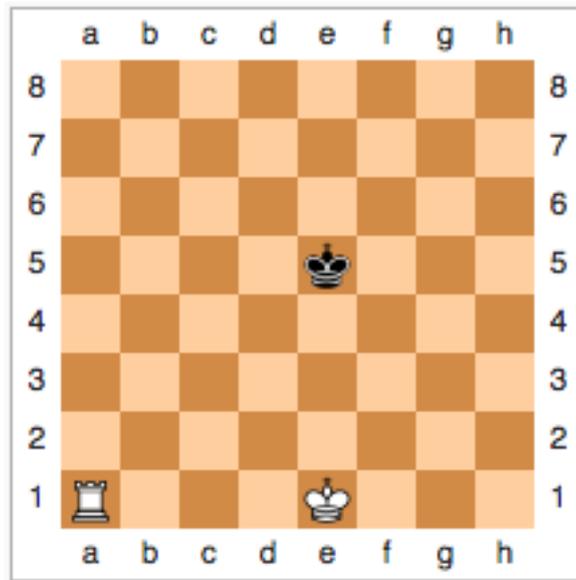


Figura 5: Esempio di finale Re e Torre contro Re.

- una funzione che gestisca tutto il ciclo della partita fino alla sua conclusione, in modo che una volta effettuata la mossa da parte dell'utente, il computer esegua la sua visualizzando la posizione attuale;
- una funzione che esegue la mossa per il nero (CPU). A voi la scelta se casuale o no.

Di seguito viene presentato un esempio di come il gioco potrebbe visualizzarsi sulla console:

```
Nuova partita: Tu (BIANCO) vs CPU (nero)

- Devi riuscire a dare scacco matto
  in massimo 50 mosse (100 semimosse) -

** Tocca al BIANCO **

8 . . . . .
7 . . . . .
6 . . . . .
5 . . . . r . . . .
4 . . . . .
3 . . . . .
2 . . . . .
1 R . . . . . T
  a b c d e f g h

Tocca a te.
Inserire mossa (es. e2e4):h1h4
Il bianco ha mosso!

** Tocca al nero **

8 . . . . .
7 . . . . .
6 . . . . .
5 . . . . r . . . .
4 . . . . . T
3 . . . . .
2 . . . . .
1 R . . . . .
  a b c d e f g h
```

Figura 6: Esempio di gioco sulla console.