Laboratorio di Calcolo per Fisici, Ottava esercitazione

Canale D-K, Docente: Dott.ssa Livia Soffi,
Esercitatori: Prof. S. Rahatlou e Dott.ssa Giulia D'Imperio

Lo scopo dell'ottava esercitazione è imparare ad utilizzare le funzioni in C.

La *morra cinese* è un antico gioco che si svolge tra due persone. Ciascuno dei giocatori porta la mano rapidamente davanti a sé facendo un gesto convenzionale (*getto*). I gesti possono essere: sasso, carta e forbici. Si confrontano i segni di ciascun giocatore e si stabilisce chi vince secondo le regole:

- Il sasso vince sulle forbici.
- Le forbici vincono sulla carta.
- La carta vince sul sasso.

Due segni uguali conducono al pareggio e il risultato viene ignorato. Vince la partita chi, dopo un numero prestabilito di *giocate*, ottiene il maggior numero di vittorie.

► **Prima parte** Scrivere un programma **morra.c** che simuli la *morra cinese*.

Il programma **morra.c** dovrà:

1. Stampare un messaggio iniziale che spiega che cosa fa (max 1 riga);
2. Utilizzando una funzione **getN** di tipo intero, chiedere all'utente di quante *giocate* (n) sarà composta la partita. Controllare che n sia positivo e minore o uguale a 20. In caso contrario stampare un messaggio di errore e chiedere di nuovo l' input.
3. Per ogni *giocata*:
   - Simulare il *getto* di ciascuno dei due giocatori. Per farlo usare una funzione **getto** di tipo int che, utilizzando i numeri casuali, restituisca con uguale probabilità uno dei tre gesti (sasso, forbici, carta). Chiamare la funzione **getto** due volte (una per il giocatore A, una per il giocatore B).
   - Decidere chi ha vinto la mano con una funzione **decido** che determina, in base alle regole sopra citate, se ha vinto A, ha vinto B, o il *getto* è finito in parità, restituendo un valore che permetta di discriminare fra i tre casi.

- Assegnare un punto al vincitore e zero al perdente, o zero a tutti e due in caso di pareggio;
- Stampare su terminale un indice che idenfichi il numero della giocata in corso e i punteggi parziali dei due giocatori (somma dei punti fino a quel momento ottenuti da ciascun giocatore).

4. Decretare il vincitore della partita e stampare su schermo un messaggio di annuncio insieme ai punteggi finali dei giocatori A e B

► **Seconda parte** Graficare i punteggi della morra cinese. Per farlo:

- Scrivere su un file **punteggio.dat** l'indice della giocata e le serie temporali dei punteggi dei due giocatori calcolati giocata dopo giocata.
- Scrivere uno script python chiamato **morra.py** il cui risultato sia quello di creare un unico grafico **punteggio.png** contenente le serie temporali dei punteggi dei due giocatori per una partita da 20 giocate.

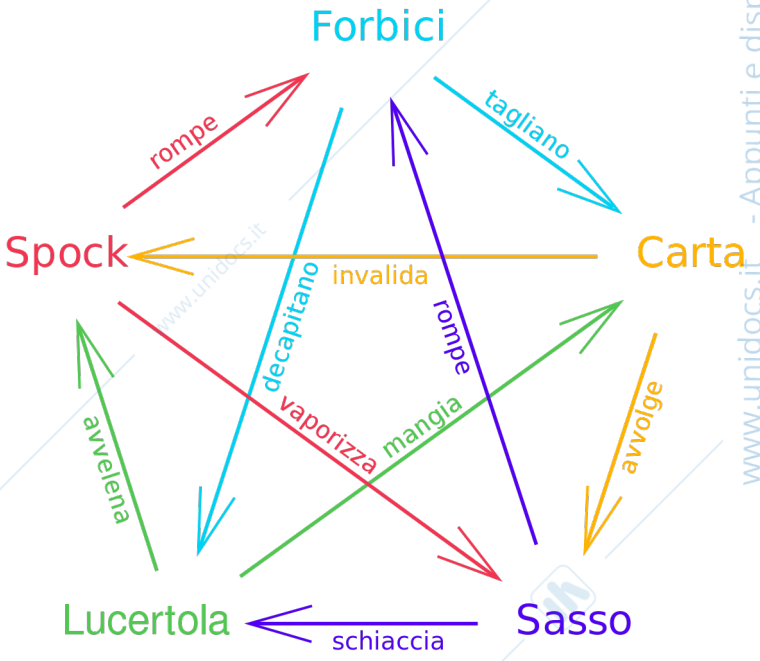
Laboratorio di Calcolo per Fisici, Ottava esercitazione - Addendum

Canale D-K, Docente: Dott.ssa Livia Soffi,
Esercitatori: Prof. S. Rahatlou e Dott.ssa Giulia D'Imperio

► Terza parte

Sasso-carta-forbice-lizard-Spock (Rock-paper-scissors-lizard-Spock) è una variante fantasiosa inventata da due studenti americani, Sam Kass e Karen Bryla, e resa successivamente famosa nell'ottavo episodio della seconda stagione del telefilm The Big Bang Theory.
La variazione consiste nell'introduzione di due nuovi segni, la lucertola (in inglese lizard) e "Spock" (celebre personaggio della serie TV Star Trek) che aumentano le combinazioni possibili nel tentativo di diminuire i pareggi nel gioco della *morra cinese*.
Scrivere un programma **lizard.c**, analogo al precedente, e che ne ripeta tutti i passi da 1 a 9, ma che simuli la variante della morra cinese in esame, rispettando le seguenti combinazioni:

- Le forbici decapitano la lucertola
- La lucertola mangia la carta
- La carta smentisce Spock
- Spock vaporizza il sasso
- Il sasso schiaccia la lucertola
- La lucertola avvelena Spock
- Spock rompe le forbici
- Le forbici tagliano la carta
- La carta avvolge il sasso
- Il sasso rompe le forbici



Eseguire i due programmi con numero di getti pari a 20 e confrontare il numero di pareggi ottenuti nei due casi.

PROGRAMMA ESEGUIBILE IN C (morra.c)

```
#include<stdio.h>
#include<math.h>
#include<time.h>
#include<stdlib.h>

int getN();
int getto();
int decido(int c, int d);

//FUNZIONE main
int main (){
int i, n;
int seed;
int gameA, gameB, result;
int ptA=0, ptB=0, pareggi=0;

seed=time(0);
srand(seed); //inizializzo l'algoritmo di generazione

 printf("\nBenvenuto! Questo programma simula la morra cinese.\n");
n = getN();

 for(i=0; i<n; i++){
 gameA = getto();
 gameB = getto();
 result = decido(gameA, gameB);

 if(result==1) {
  ptA++;
  printf("--------------------GIOCATA %d.--------------------\nVince A!\n", i+1);
}
else if(result==2) {
  ptB++;
 printf("--------------------GIOCATA %d.--------------------\nVince B!\n", i+1);
 }
else {
 printf("--------------------GIOCATA %d.--------------------\nParità.\n", i+1);
pareggi++;
 }

printf("PUNTI A=%d;\t\tPUNTI B=%d.\n", ptA, ptB);
 }

 if(ptA>ptB) printf("\nLa partita è terminata. Vince A con %d punti!\n", ptA);
 else if(ptA<ptB) printf("\nLa partita è terminata. Vince B con %d punti!\n", ptB);
 else if(ptA==ptB) printf("\nLa partita è terminata in parità.\n");

printf("Questa partita è terminata con %d pareggi.\n\n", pareggi);

 return 0;
}


//FUNZIONE getN
int getN(){
  int a;
```

```
 printf("Inserisci un numero di giocate minore di 20.\n#giocate: ");

 do{
   scanf("%d", &a);
   if(a<=0 || a>20) printf("Errore. Inserisci un valore valido.\n#giocate: ");
 } while(a<=0 || a>20);

 return a;
}


//FUNZIONE getto
int getto(){
 int b;

b = rand()%3;

 return b;
}


//FUNZIONE decido
int decido(int c, int d){
  int e=0;

 if(c==d) { e=0; }
 else if((c==0 && d==1) || (c==1 && d==2) || (c==2 && d==0)) { e=1; }
 else { e=2; }

 return e;
}
```

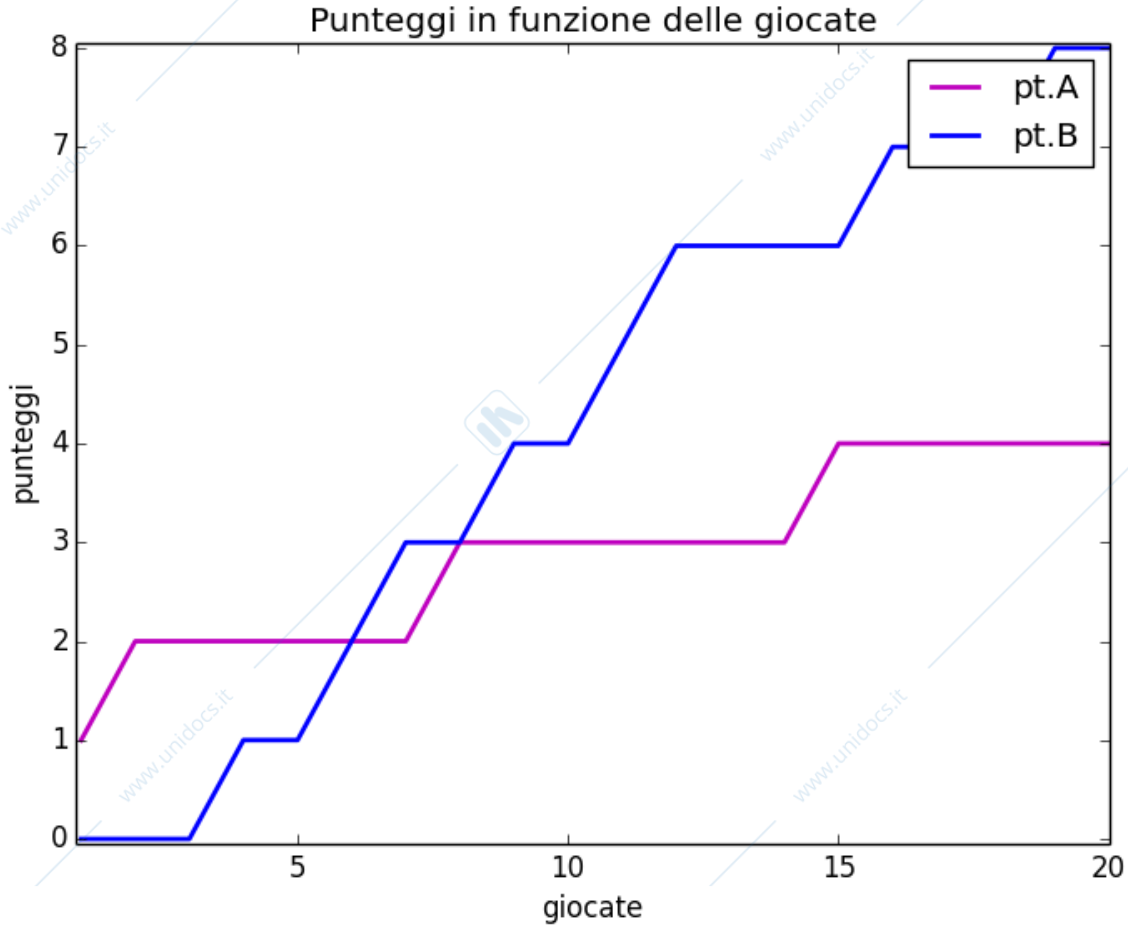
FILE (punteggio.dat) UTILIZZATO

```
#giocata #A  #B
01       1   0
02       2   0
03       2   0
04       2   1
05       2   1
06       2   2
07       2   3
08       3   3
09       3   4
10       3   4
11       3   5
12       3   6
13       3   6
14       3   6
15       4   6
16       4   7
17       4   7
18       4   7
19       4   8
20       4   8
```

PROGRAMMA IN PYTHON (morra.py)

```
#This Python file uses the following encoding: utf-8

import matplotlib.pyplot as plt
import numpy as np

x,y=np.loadtxt('punteggio.dat', unpack=True, usecols=(0,1))
plt.title('Punteggi in funzione delle giocate')
plt.xlim(0.9, 20.01)
plt.ylim(-0.05, 8.05)
plt.xlabel('giocate')
plt.ylabel('punteggi')
plt.plot(x,y,'m-', label='pt.A', linewidth=2)
plt.legend()

x,y=np.loadtxt('punteggio.dat', unpack=True, usecols=(0,2))
plt.plot(x,y,'b-', label='pt.B', linewidth=2)
plt.legend()

plt.savefig('punteggio.png')
plt.show()
```

ALLEGATO DI COME TI DOVREBBE USCIRE IL GRAFICO (punteggio.png)



ADDENDUM

FILE ESEGUIBILE IN C (lizard.c)

```
#include<stdio.h>
#include<math.h>
#include<time.h>
#include<stdlib.h>

int getN();
int getto();
int decido(int c, int d);

//FUNZIONE main
int main (){
int i, n;
int seed;
int gameA, gameB, result;
int ptA=0, ptB=0, pareggi=0;

seed=time(0);
srand(seed); //inizializzo l'algoritmo di generazione

printf("\nBenvenuto! Questo programma simula una fantasiosa variante della morra cinese
        che ne minimizzi i pareggi.\n");
 n = getN();

for(i=0; i<n; i++){
 gameA = getto();
 gameB = getto();
result = decido(gameA, gameB);

 if(result==1) {
  ptA++;
 printf("--------------------GIOCATA %d.--------------------\nVince A!\n", i+1);
  }
 else if(result==2) {
ptB++;
printf("--------------------GIOCATA %d.--------------------\nVince B!\n", i+1);
 }
 else {
  printf("--------------------GIOCATA %d.--------------------\nParità.\n", i+1);
  pareggi++;
 }

printf("PUNTI A=%d;\t\tPUNTI B=%d.\n", ptA, ptB);
 }

 if(ptA>ptB) printf("\nLa partita è terminata. Vince A con %d punti!\n", ptA);
 else if(ptA<ptB) printf("\nLa partita è terminata. Vince B con %d punti!\n", ptB);
 else if(ptA==ptB) printf("\nLa partita è terminata in parità.\n");

printf("Questa partita è terminata con %d pareggi.\n\n", pareggi);

return 0;
}
```

```
//FUNZIONE getN
int getN(){
 int a;

 printf("Inserisci un numero di giocate minore di 20.\n#giocate: ");

do{
  scanf("%d", &a);
 if(a<=0 || a>20) printf("Errore. Inserisci un valore valido.\n#giocate: ");
 } while(a<=0 || a>20);

return a;
}


//FUNZIONE getto
int getto(){
 int b;

 b = rand()%5;

return b;
}


//FUNZIONE decido
int decido(int c, int d){
 int e;

 if(c==d) { e=0; }
else if((c==0 && (d==1 || d==2)) || (c==1 && (d==2 || d==3)) || (c==2 && (d==3 || d==4)) ||
(c==3 && (d==4 || d==0)) || (c==4 && (d==0 || d==1))) { e=1; }
  else { e=2; }

return e;
}
```

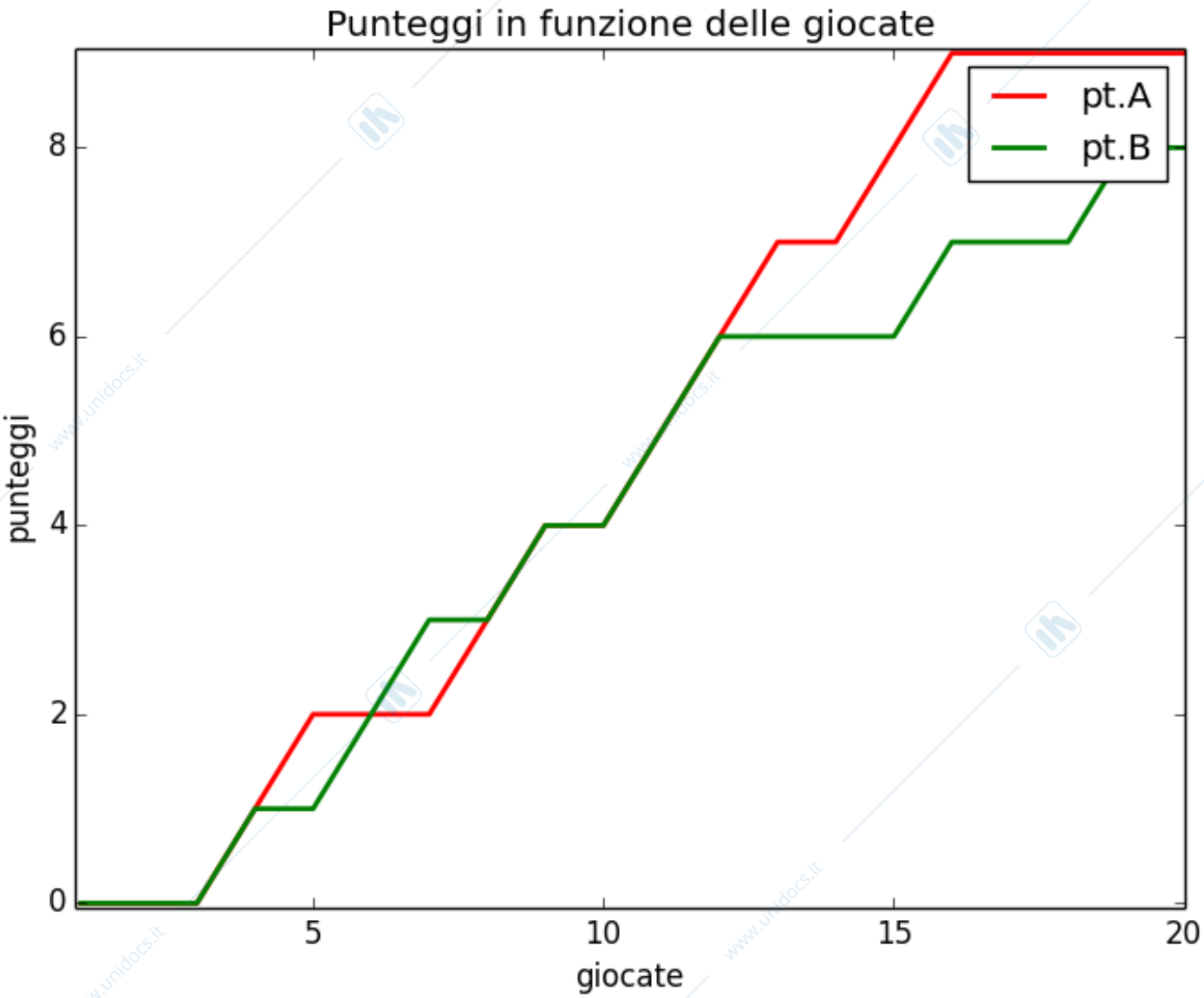
FILE (ptLizard.dat) UTILIZZATO

```
#giocata #A  #B
01       0   1
02       0   1
03       0   2
04       1   2
05       2   2
06       2   3
07       2   4
08       3   4
09       4   4
10       4   5
11       5   5
12       6   5
13       7   5
14       7   6
15       8   6
16       9   6
17       9   7
18       9   7
19       9   8
20       9   9
```

PROGRAMMA IN PYTHON (lizard.py)

```
#This Python file uses the following encoding: utf-8

import matplotlib.pyplot as plt
import numpy as np

x,y=np.loadtxt('ptLizard.dat', unpack=True, usecols=(0,1))
plt.title('Punteggi in funzione delle giocate')
plt.xlim(0.9, 20.01)
plt.ylim(-0.05, 9.05)
plt.xlabel('giocate')
plt.ylabel('punteggi')
plt.plot(x,y,'r-', label='pt.A', linewidth=2)
plt.legend()

x,y=np.loadtxt('punteggio.dat', unpack=True, usecols=(0,2))
plt.plot(x,y,'g-', label='pt.B', linewidth=2)
plt.legend()

plt.savefig('lizard.png')
plt.show()
```

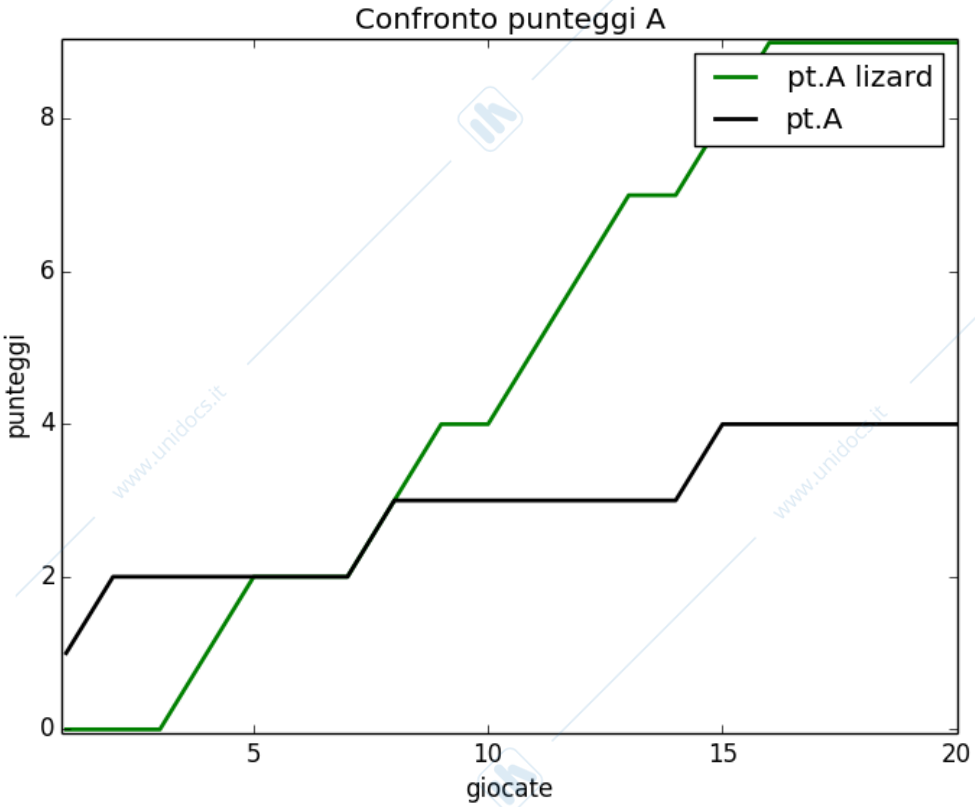
ALLEGATO DI COME TI DOVREBBE USCIRE IL GRAFICO (lizard.png)



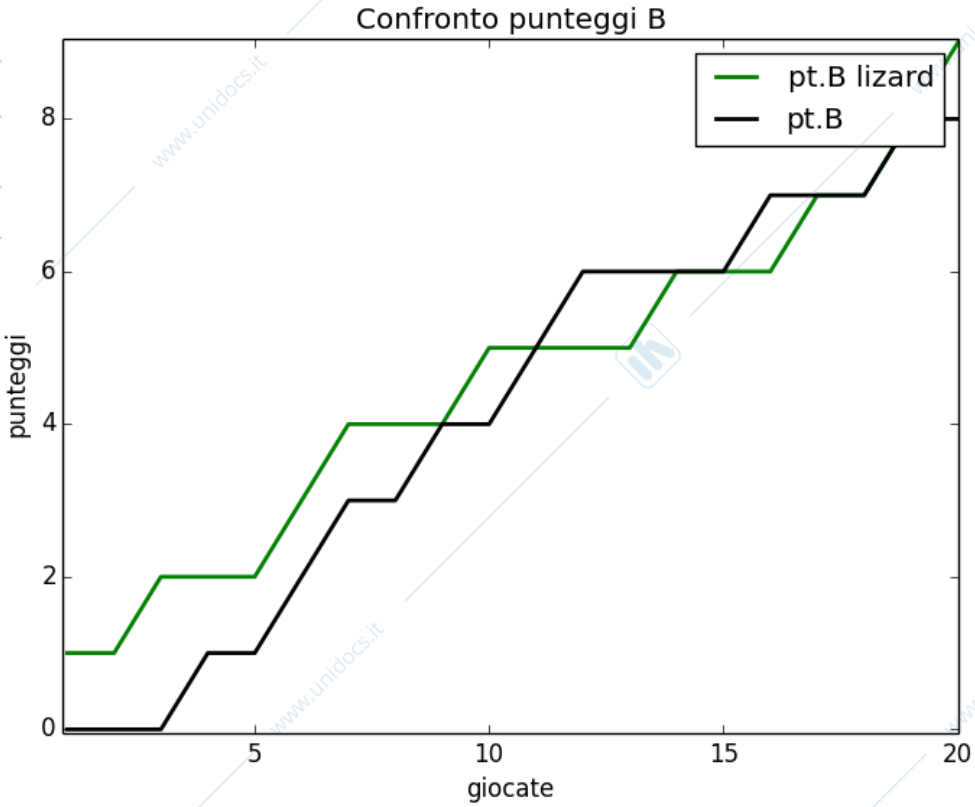
PROGRAMMA IN PYTHON: CONFRONTI FRA I PUNTEGGI CON "METODO SHELDON" E MORRA NORMALE (confronti.py)

```
#This Python file uses the following encoding: utf-8

import matplotlib.pyplot as plt
import numpy as np

#primo grafico: confronti di A
x,y=np.loadtxt('ptLizard.dat', unpack=True, usecols=(0,1))
plt.title('Confronto punteggi A')
plt.xlim(0.9, 20.01)
plt.ylim(-0.05, 9.05)
plt.xlabel('giocate')
plt.ylabel('punteggi')
plt.plot(x,y,'g-', label='pt.A lizard', linewidth=2)
plt.legend()

x,y=np.loadtxt('punteggio.dat', unpack=True, usecols=(0,1))
plt.plot(x,y,'k-', label='pt.A', linewidth=2)
plt.legend()

plt.savefig('confrontiA.png')
plt.show()

#secondo grafico: confronti di B
x,y=np.loadtxt('ptLizard.dat', unpack=True, usecols=(0,2))
plt.title('Confronto punteggi B')
plt.xlim(0.9, 20.01)
plt.ylim(-0.05, 9.05)
plt.xlabel('giocate')
plt.ylabel('punteggi')
plt.plot(x,y,'g-', label='pt.B lizard', linewidth=2)
plt.legend()

x,y=np.loadtxt('punteggio.dat', unpack=True, usecols=(0,2))
plt.plot(x,y,'k-', label='pt.B', linewidth=2)
plt.legend()

plt.savefig('confrontiB.png')
plt.show()
```

ALLEGATI DI COME TI DOVREBBERO USCIRE I GRAFICI



(confrontiA.png)



(confrontiB.png)