Exercise 1

1. Probability space:
Suit $= \{\clubsuit, \diamondsuit, \heartsuit, \spadesuit\}$
Number $= \{Ace, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, King\}$
$\Omega = \{(Ace, \clubsuit), (2, \clubsuit), (3, \clubsuit), (4, \clubsuit), (5, \clubsuit), (6, \clubsuit)(7, \clubsuit),$
$(8, \clubsuit), (9, \clubsuit), (10, \clubsuit), (Jack, \clubsuit), (Queen, \clubsuit), (King, \clubsuit), ...\}$
where each element in Ω can be represented as picking a card formed by a pair (number, suit).
Each element in Ω has probability $\frac{1}{52}$.

2a. The first two cards include at least one Ace:
We can define the probability of the event E, "pick at least one ace", like $1-$ the probability of getting two cards different from an Ace.
$Pr(E) = 1 - \dfrac{48}{52}\dfrac{47}{51} = 1 - 0.85 = 0.15$

2b. The first five cards include at least one Ace
We have to take into account for drawing 1, 2, 3 or 4 aces. The probability of the event E, "at least one Ace", is:
$Pr(E) = 1 - \dfrac{\binom{48}{5}}{\binom{52}{5}} = 0.35$

2c. The first two cards are a pair of the same rank:
We define the event E = "pair", and its probability will be:
$P(E) = \dfrac{13 \cdot \binom{4}{2}}{\binom{52}{2}} = 0.058$

2d. The first five cards are all diamonds:
We again call this event E:
$Pr(E) = \dfrac{\binom{13}{5}}{\binom{52}{5}} = 0.00049$

2e. The first five cards form a full house:
We can model this event E as follows: the total number of full houses over the total number of permutations of 52 cards over 5 positions. The total number of full houses, FH, is:
$FH = 13\binom{4}{3} \cdot 12\binom{4}{2} = 3744$
$Pr(E) = \dfrac{3744}{\binom{52}{5}} = 0.0014$
All these cases were implemented in Python (see *ex1DM.py* file). The file can be executed from the terminal with the command: *python ex1DM.py*. The used version of Python is 2.7.
In fact, after many draws the results are pretty much the same. For the case e there were asked 100.000 draws in order to observe some successful cases of having a served full house.

Exercise 2

1. Sample space

The sample space Ω is the set of all possible outcomes of the random process modeled by the probability space. In this case Ω is represented by all the possible configurations of n men and m women around a round table. For example, if $n = 5$ and $m = 3$, then a possible configuration could be: {man, woman, woman, man, man, man, woman, man}. In this case, the number of men sitted next to at least a woman is 4. Please notice that with the particular case of a round table, the first position in the example above is to be considered also next to the last position, i.e the permutation is circular. In order to be more detailed, the number of all the possible permutations of men and women around the table is $(n+m)!$, each one with probability $\frac{1}{(n+m)!}$.

In order to be specific, $\Omega = \{(man, woman, man, woman, man, man, man),$
$(woman, woman, man, man, man, man, man)...\}$, where the $n = 5$ and $m = 2$.

In an another way, one could specify Ω as the set of all possible ways in which a man can be sitted to a table, with particular attention to the people he has next to him. For example, a man can be sitted near two other men, near two other women, or a combination of the two. In this case, Ω would be:
$\Omega = \{(man, man, man), (woman, man, man)$
$, (man, man, woman), (woman, man, woman)\}$.
With this definition of the sample space, Ω size will be 4. Notice that we are considering only the man's positioning.

2. Expected number of men sitted next to at least a woman

A random variable $\boldsymbol{X}$ on a sample space Ω is a function on Ω, that is: $X : \Omega \to \Re$
The expectation of a discrete random variable $\boldsymbol{X}$, denoted by $\mathbf{E[X]}$. It is given by : $E[X] = \sum_i iPr(X = i)$
Let $\mathbf{X}$ be "*the number of men sitted next to at least a woman*". We define N 0,1 random variables X_i:

$$x_i = \begin{cases} 1\text{, if man i seats near at least a woman} \\ 0\text{, otherwise} \end{cases}$$

$E[X_i] = 1 \cdot Pr(X_i = 1) + 0 \cdot Pr(X_i = 0)$
$Pr(X_i = 1) = 1 - \frac{\binom{n-1}{2}}{\binom{m+n-1}{2}}$
So the probability a man is sitted near at least a woman is given by the difference between 1 and the probability of that man to sit near two men.
$E[X] = 2 + \sum_{n=3}^{N} E[X_i] = \sum_{n=1}^{N} 1 - \frac{\binom{n-1}{2}}{\binom{n+m-1}{2}}$
The number 2 is due to the fact that with 1 man or 2, and $m > 1$, then the probability that both men sit next to a woman is 1. This is why the sum starts with n=3. As an example, if m=10 and n=10, the expected value of men sitted next to at least a woman is 8.

Exercise 3

1. Probability space

In a graph $G_{(n,p)}$, where n is the number of nodes and p the probability that an edge between any two nodes of the graph exists, we can represent Ω as the set of all possible nodes between the n nodes belonging to the graph. We can therefore define an edge e between two nodes x and y as a tuple $edge = (x, y)$.
Therefore:
$\Omega = \{e_{1,2}, e_{1,3}, e_{1,4}, ..., e_{1,n}, e_{2,3}, e_{2,4}, ..., e_{2,n}, ...e_{n-1,n}\}$
The size of the sample space will be:
$\mid \Omega \mid = \sum_{i=1}^{n} n - i$, where n is the number of nodes of the model.

2. What is the probability of each element in Ω?

Since in this kind of model we have that an edge exists with probability p, then each element in Ω has probability p.

What is the probability that the graph contains exactly two cycles of length $\frac{n}{2}$?

For solving this question, we can divide the entire probability as the product of two probabilities, of two distinct events. The first event is representing the probability of having a cycle over $\frac{n}{2}$ nodes, and no other edges between other nodes. Let's call this event *Cycle1*, so its probability is defined as follows:

$$Pr(Cycle1) = \frac{\binom{n}{n/2}(n/2-1)!p^{n/2}(1-p)^{\frac{n(n-1)}{2}-\frac{n}{2}}}{n!p^{\frac{n(n-1)}{2}}}$$

$\binom{n}{n/2}$ is the number of all the possible modes of getting $\frac{n}{2}$ nodes from n, multiplied by the number of possible orderings of the nodes $(n/2-1)!$, multiplied by the probability of these edges of existing, all this being multiplied by the probability of not having the other edges among the ones not belonging to the cycle, that is $(1-p)^{\frac{n(n-1)}{2}-\frac{n}{2}}$. At the denominator we have all the possible permutations of the n nodes, multiplied by the probability of each possible node between these n nodes.

Then we call another event *Cycle2*, that also must be long $\frac{n}{2}$ and built over the remaining nodes because of the first cycle. The reasoning is the same, with the only two differences that the value of the binomial coefficient $\binom{n/2}{n/2}$ is one, and the exponent of $(1-p)$ is different, because we have to take into consideration the nodes already *used* for constructing the first cycle. Therefore the probability of the second cycle will be:

$$Pr(Cycle2) = \frac{\binom{n/2}{n/2}(n/2-1)!p^{n/2}(1-p)^{\frac{(n-n/2)(n-n/2-1)}{2}-\frac{n}{2}}}{\frac{n}{2}!p^{\frac{(n-n/2)(n-n/2-1)}{2}}}$$

In conclusion, the probability of having exactly two cycles of length $\frac{n}{2}$ will be the product: $Pr(Cycle1) \cdot Pr(Cycle2)$.

4. What is the probability of having exactly two cycles of any length, and no other edge?
We can consider this case as a generalization of the previous question, in which the two cycles can have arbitrary lengths, let's say k and j. We will do the same reasoning, first considering the cycle of length k, then the cycle of length j, and the final probability will be the product of the two.

$$Pr(CycleK) = \frac{\binom{n}{k}(k-1)!p^k(1-p)^{\frac{n(n-1)}{2}-k}}{n!p^{\frac{n(n-1)}{2}}}$$

$$Pr(CycleJ) = \frac{\binom{n-k}{j}(j-1)!p^j(1-p)^{\frac{(n-k)(n-k-1)}{2}-j}}{(n-k)!p^{\frac{(n-k)(n-k-1)}{2}}}$$

The differences between the *Cycle2* of the previous point and the *CycleJ* of this point are that the binomial coefficient is not equal to one in this case, because we are not anymore considering two cycles of length $\frac{n}{2}$, and the exponents, that are influenced by k instead of $\frac{n}{2}$.

5. What is the expected degree of a node?
Each node can be connected to other $n-1$ nodes through n-1 edges, but these edges must exist, so $n-1$ is multiplied by the probability p. Therefore, the average node degree will be $p(n-1)$.

6. What is the expected number of edges?
We represent as X the vector of Bernoulli random variables representing the edges, each edge exists with probability p, therefore the expected number S_n of expected edges will be:
$E[S_n] = E\sum_i^{\binom{n}{2}} x_i = p \cdot \binom{n}{2}$.

7. What is the expected number of houses subgraphs?
We first define t as the number of possible subsets of 5 nodes that are possible from n, $t = \binom{n}{5}$, then we define t random variables as follows: $X = \{X_1, X_2, ..., X_t\}$, where:

$$x_i = \begin{cases} 1, \text{ if the subset forms a house subgraph} \\ 0, \text{ otherwise} \end{cases}$$

Now we define $E[X_i] = 1Pr(X_i = 1) + 0Pr(X_i = 0)$, and we define the probability of a defined variable to be a house like:
$Pr(X_i = 1) = \frac{5!}{\binom{10}{5}}p^6(1-p)^4$.
This represents the probability of a subset of 5 nodes that can form a house subgraph. More in details, this probability is computed multiplying the probability of taking 6 edges, p^6, 5 from the *perimeter* of the house, plus one internal to it, by the probability of not taking any of the other 4 remaining edges, $(1-p)^4$(in

fact, with 5 nodes we can have at most 10 undirected edges). All this is multiplied by all the possible permutations of the 5 nodes, that represent all the ways in which nodes can be *arranged* in order to form a house. Finally, at the denominator we have the number of modes in which we can choose 5 edges from a set of 10.
Knowing that the general expected value can be computed as:
$E[X] = \sum_{t=1}^{t} E[X_i]$, we can declare that the expected number of house subgraphs in a random graph $G_{n,p}$ is:
$E[X] = \binom{n}{5} \frac{5!}{\binom{10}{5}} p^6 (1-p)^4$

Exercise 4

For this exercise three commands were used, awk, sort and head. With awk command, since we were not interested in the score of the review, we just count the number of reviews for the same type of beer, using a dictionary provided by awk. At the end of the scrolling, the number of reviews and the type of beers were written to the output, given as input to the sort command. Then the file was sorted reversely according to the number of reviews, and then head outputs on the terminal the first 10 lines.
The command is:
{awk '{$NF=””; a[$0]+=1} END {for (i in a) print a[i], i}}' beers.txt | sort -k1 -n -r | head

```
MacBook-Pro-di-Ciprian:Desktop mychro94$ awk '{$NF=""; a[$0]+=1} END {for (i in a) print a[i], i}' beers.txt | sort -k1 -n -r | head
3696 Guinness Draught
3662 Pabst Blue Ribbon
3230 Dogfish Head 90 Minute Imperial IPA
3126 Budweiser
3119 Sierra Nevada Pale Ale &#40;Bottle&#41;
3110 Samuel Adams Boston Lager
3056 Chimay Bleue &#40;Blue&#41; / Grande Rserve
2904 North Coast Old Rasputin Russian Imperial Stout
2872 Stone Arrogant Bastard Ale
2813 Orval
```

Exercise 5

For this exercise the job to be done was similar as in exercise 4, with the difference that we cared about also about the score of the review. Again a dictionary was used, in which the key was the beer, and the value a tuple (total score, number of reviews).
See *ex5DM.py* file. It can be run from terminal with the command *python ex5DM.py*. The folder must contain the file *beers.txt*.

```
MBP-di-Ciprian:python mychro94$ python ex5DM.py
number of different beers: 110298
Top 10 beers
Westvleteren 12 with average: 18
Lost Abbey Veritas 004 with average: 17
De Dolle Speciaal Brouwsel 20 with average: 17
Stone Imperial Russian Stout with average: 17
Russian River Deviation &#40;Bottleworks IX&#41; with average: 17
Three Floyds Vanilla Bean Barrel Aged Dark Lord Russian Imperial Stout with aver
age: 17
3 Fonteinen Oude Geuze 1998 &#40;50th Anniversary&#41; with average: 17
Westvleteren Extra 8 with average: 17
Cigar City Hunahpus Imperial Stout - Bourbon Barrel Aged with average: 17
Founders KBS &#40;Kentucky Breakfast Stout&#41; with average: 17
MBP-di-Ciprian:python mychro94$
```

Exercise 6

The program is making a first request to the kijiji site in order to find the total number of pages to be visited. Then, all the html text not regarding the job announcements is thrown away, and a search for all *li* elements is done. Each *li* element contains all the informations about a job announcement. So from each *li*, title, url, description, place and publication date are extracted and written to the output file.
Libraries *requests*, *BeautifulSoup* and *re* were used. See *ex6DM.py* file for the code. The output will be in the *jobs.txt* file.

```
jobs.txt
1) Laureato esperto Java        Il gruppo Wish per il potenziamento della propria struttura e del proprio team, cerca laureato:

Analista Programmatore JAVA
Requisiti tecnici:

Java
J2EE
Angular
Java Framework quali Spring, Struts, Spring MVC
DBMS quali Oracle, MySQL, PostgreSQL
HTML5, CSS3, XML, Javascript
Web Services
Compenso mensile base di 3000 euro netti + incentivi

Sede lavorativa: FIRENZE...     Firenze Oggi, 00:08     https://www.kijiji.it/annunci/offerta/firenze-annunci-firenze/laureato-esperto-java/130852163
2) Assumiamo Sviluppatore PHP a tempo indeterminato a Milanofiori       Buongiorno, il nostro gruppo sta selezionando sviluppatori PHP.

Offriamo un ambiente tecnologicamente stimolante in cui intraprendere un percorso professionale volto ad accrescere le competenze tecniche, di seguito le caratteristiche
dell'offerta:
* Sede di lavoro: Centro Direzionale Milanofiori
* Disponibilità Full Time
* Contratto a tempo indeterminato 4 livello CCNL Metalmeccanico e Industria
* Buoni pasto, smartworking, programmi di welfare aziendale

Competenze tecniche richieste:
* Buona conoscenza del linguaggio PHP (sviluppo back-end);
* Buona conoscenza di Mysql
* Buona conoscenza di HTML5, CSS3, Javascript
* Buona conoscenza di Jquery e Bootstrap
* Buona capacità di programmare nella logica Object-Oriented;
* Conoscenza ed utilizzo di git


Competenze tecniche preferenziali:
* Conoscenza di C e C++
* Programmazione multithread
* Competenze sistemistiche in ambiente Linux


Caratteristiche del candidato/a:
* Diploma
* Laurea triennale preferenziale
* Buona conoscenza della lingua inglese (almeno livello B2)
* Età preferenziale 27-35 anni
* Almeno 2-3 anni di esperienza lavorativa...   ~Altre zone     Ieri, 11:20     https://www.kijiji.it/annunci/offerta/milano-annunci-altre-zone/assumiamo-sviluppatore-php-a-
tempo-indeterminato-a/130870144
3) Firmware Developer ambito R&D        MOVIA SpA opera da oltre un decennio nel mercato italiano in cui si è velocemente affermata come nuova realtà strutturata e
competitiva, proponendosi come fornitore dinamico e flessibile, in grado di offrire un'ampia ed articolata gamma di Servizi e Soluzioni ICT: dall'Outsourcing al Facility
Management, dal Desktop Management al System & Network Management.
Movia SpA è oggi una realtà in continua ed importante crescita ed espansione, che crede ed investe costantemente sulle Persone.


In questo momento siamo alla ricerca di un:
```