

Task 1

Your task is to construct following CSV formatted list from provided data (variables bellow - **html_data** and **json_data**):

```
name,age,origin,occupation
Joana Harjo,23,Mars,pilot
Chery Auvil,23,Pluto,pilot
Doris Walrath,29,Jupiter,researcher
Jasmine Botello,24,Saturn,researcher
Denisha Perera,23,Uranus,leader
Laurel Altizer,25,Neptun,researcher
Wynell Granda,25,Neptun,pilot
Lester Delao,22,Jupiter,pilot
Kaleigh Hamlin,24,Mars,engineer
Alonzo Zukowski,29,Earth,engineer
Odette Pursell,20,Earth,engineer
Willene Merck,29,Mars,pilot
Mui Delbosque,26,Mars,engineer
Rickie Selman,25,Earth,researcher
```

Detail instructions:

1. get list of **people and their attributes** from table rows (html_data)
2. find who has **occupation** and who has not
3. **print only** the persons who have occupation and their attribute alive equals to Yes.

Notes:

- you should not copy/paste any part of input data. If the source data change, also the output of your code should change accordingly
- probably the most easy solution is via libraries: **BeautifulSoup** and **json**. However feel free to use any other suitable library.
- CSV format - comma separated values, new line for every record

html_data = ""

```
<table class="tg"> <tr> <th class="tg-031e">Name</th> <th class="tg-031e">Age</th> <th
class="tg-031e">Origin</th> <th class="tg-yw4l">Alive</th> </tr> <tr> <td
class="tg-031e">Joana Harjo</td> <td class="tg-031e">23</td> <td
class="tg-031e">Mars</td> <td class="tg-yw4l">Yes</td> </tr> <tr> <td
class="tg-031e">Chery Auvil</td> <td class="tg-031e">23</td> <td
class="tg-031e">Pluto</td> <td class="tg-yw4l">Yes</td> </tr> <tr> <td
class="tg-huad">Cecily Soden</td> <td class="tg-huad">27</td> <td
class="tg-huad">Earth</td> <td class="tg-i6eq">No</td> </tr> <tr> <td
class="tg-031e">Doris Walrath</td> <td class="tg-031e">29</td> <td
class="tg-031e">Jupiter</td> <td class="tg-yw4l">Yes</td> </tr> <tr> <td
class="tg-031e">Jasmine Botello</td> <td class="tg-031e">24</td> <td
```

```

class="tg-031e">Saturn</td> <td class="tg-yw4l">Yes</td> </tr> <tr> <td
class="tg-031e">Denisha Perera</td> <td class="tg-031e">23</td> <td
class="tg-031e">Uranus</td> <td class="tg-yw4l">Yes</td> </tr> <tr> <td
class="tg-031e">Laurel Altizer</td> <td class="tg-031e">25</td> <td
class="tg-031e">Neptun</td> <td class="tg-yw4l">Yes</td> </tr> <tr> <td
class="tg-huad">Chieko Rosecrans</td> <td class="tg-huad">27</td> <td
class="tg-huad">Mars</td> <td class="tg-i6eq">No</td> </tr> <tr> <td
class="tg-031e">Wynell Granda</td> <td class="tg-031e">25</td> <td
class="tg-031e">Neptun</td> <td class="tg-yw4l">Yes</td> </tr> <tr> <td
class="tg-031e">Lester Delao</td> <td class="tg-031e">22</td> <td
class="tg-031e">Jupiter</td> <td class="tg-yw4l">Yes</td> </tr> <tr> <td
class="tg-031e">Kaleigh Hamlin</td> <td class="tg-031e">24</td> <td
class="tg-031e">Mars</td> <td class="tg-yw4l">Yes</td> </tr> <tr> <td
class="tg-031e">Alonzo Zukowski</td> <td class="tg-031e">29</td> <td
class="tg-031e">Earth</td> <td class="tg-yw4l">Yes</td> </tr> <tr> <td
class="tg-031e">Odette Pursell</td> <td class="tg-031e">20</td> <td
class="tg-031e">Earth</td> <td class="tg-yw4l">Yes</td> </tr> <tr> <td
class="tg-huad">Danial Hite</td> <td class="tg-huad">20</td> <td
class="tg-huad">Jupiter</td> <td class="tg-i6eq">No</td> </tr> <tr> <td
class="tg-huad">Clayton Earnhardt</td> <td class="tg-huad">20</td> <td
class="tg-huad">Mars</td> <td class="tg-i6eq">No</td> </tr> <tr> <td
class="tg-031e">Willene Merck</td> <td class="tg-031e">29</td> <td
class="tg-031e">Mars</td> <td class="tg-yw4l">Yes</td> </tr> <tr> <td class="tg-031e">Mui
Delbosque</td> <td class="tg-031e">26</td> <td class="tg-031e">Mars</td> <td
class="tg-yw4l">Yes</td> </tr> <tr> <td class="tg-huad">Kenya Washer</td> <td
class="tg-huad">21</td> <td class="tg-huad">Earth</td> <td class="tg-i6eq">No</td> </tr>
<tr> <td class="tg-031e">Rickie Selman</td> <td class="tg-031e">25</td> <td
class="tg-031e">Earth</td> <td class="tg-yw4l">Yes</td> </tr> <tr> <td
class="tg-031e">Marisa Gioia</td> <td class="tg-031e">24</td> <td
class="tg-031e">Earth</td> <td class="tg-yw4l">Yes</td> </tr></table>

```

```

json_data = """
{"researcher": ["Cecily Soden", "Doris Walrath", "Jasmine Botello", "Laurel Altizer", "Rickie
Selman"], "pilot": ["Joana Harjo", "Chery Auvil", "Wynell Granda", "Lester Delao", "Willene
Merck"], "engineer": ["Kaleigh Hamlin", "Alonzo Zukowski", "Odette Pursell", "Danial Hite",
"Clayton Earnhardt", "Mui Delbosque", "Kenya Washer"], "leader": ["Denisha Perera",
"Chieko Rosecrans"]}
"""

```

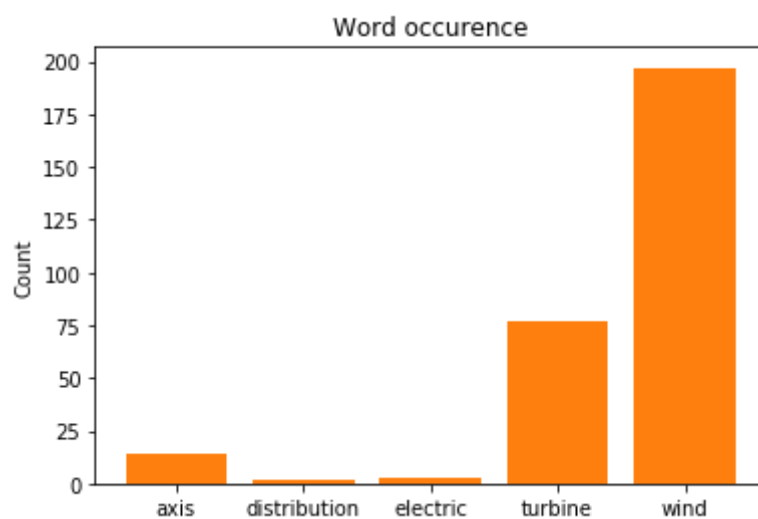
Task 2

Obtain the figure below. Detail instructions:

1. download html content from wikipedia page provided variable `url`
2. search for words (list `words`) in the html content and count their occurrence
3. plot the occurrence in bar chart

Notes:

- you can ignore cleaning of html, converting to lowercase etc. Actual count do NOT need to be exactly the same as provided in figure.



`url` = "https://en.wikipedia.org/wiki/Wind_turbine"

`words` = ["turbine", "wind", "axis", "distribution", "electric"]

Task 3

The time serie is defined by variables *time* and *data* which are generated by following code:

```
import numpy as np
### input data
N = 600
a = 0.2
b = 0.8
c = 0.9
d = 20
h = 10.0
data = np.zeros(N)
data[0] = 0.1
for k in range(0,N-1):
    data[k+1] = (c*data[k]) + ( (a*data[k-d]) / (b + (
data[k-d]**h)))
time = np.arange(0, 600) / 10.
```

*** Just copy-paste this code snippet to your task solution source code.**

- 1) Plot the variable *data* (axis y) against the variable *time* (axis x) to a figure with grid, title and well described axes.
- 2) Generate random **integer** variable *t1* from range <0; 30>
- 3) Set variable *t2* = *t1* + 10
- 4) Create a variable *data_sub* as subset of variable *data* by copying the part of *data* vector corresponding to *time* vector from *t1* to *t2*.
- 5) Add some visible noise to variable *data_sub*.
- 6) Save the *data_sub* variable with added noise to txt file.
- 7) Load the saved data from *step 6*) to a variable *data_load*.
- 8) Plot the *data_load* variable to the new figure. Use the red dashdot line.

Note: You will probably need python modules (libraries) **numpy** and **matplotlib**.

Task 4

Use **sympy** or other similar python module (library) to solve mathematical tasks below.

- 1) Determine first and second derivative of the function f with respect to x

$$f(x) = \frac{x}{\ln x}$$

- 2) Determine the partial derivatives of the function g with respect to y and z .

$$g(x) = \cos y - \ln \frac{x}{z}$$

- 3) Calculate the indefinite integral

$$\int \frac{\ln x}{-x} dx$$

- 4) Find the limit of a function below as x is going to zero.

$$\lim_{x \rightarrow 0} \frac{\sin 4x}{\sqrt{1+x}-1}$$

- 5) Find the limit of a function below as x is going to zero from **left**.

$$\lim_{x \rightarrow 0^+} x^{\frac{1}{x}}$$

- 6) Simplify the expression

$$\left(\frac{15m^3n^{-2}p^1}{25m^{-2}n^{-4}} \right)^{-3}$$