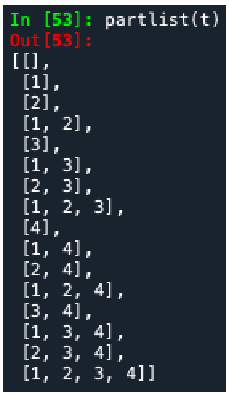
Exercises – Lecture 1
ENSEA/FAME Computer Science

Exercise 1 – The subset sum problem

Consider the following problem: given a set of integers t and an integer s, find a subset of t whose sum is equal to s. For example, if $t = \{1, 2, 3, 4\}$ and $s = 5$, then the subsets $\{1, 4\}$ and $\{2, 3\}$ are solutions to this problem. If $s = 0$, we consider that $\emptyset \subseteq t$ is a solution of the subset sum problem.
The naive method consists of testing all the subsets of t.

Question 1. Let n the number of elements of t. What is the number of subset of t ?

Question 2. Write a function `def partlist(t)` which return the list of all subset of t. For example



```
In [53]: partlist(t)
Out[53]:
[[],
 [1],
 [2],
 [1, 2],
 [3],
 [1, 3],
 [2, 3],
 [1, 2, 3],
 [4],
 [1, 4],
 [2, 4],
 [1, 2, 4],
 [3, 4],
 [1, 3, 4],
 [2, 3, 4],
 [1, 2, 3, 4]]
```

is the list of all subsets of the list $t = \{1, 2, 3, 4\}$.

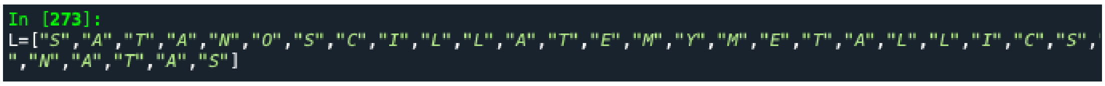
Question 3. Write a function `def subsetsum(s, t)` (whose arguments are an integer `s` and a list `t`) which return the list of all sublists of t whose sum in `s`.

Exercice 2 – Palindrome

A palindrome is a symmetrical word, it can be read from left to right and from right to left : for example

Satan oscillate my metallic sonatas

A word or a sentence will be represent by a list of letters. For the previous example, we have

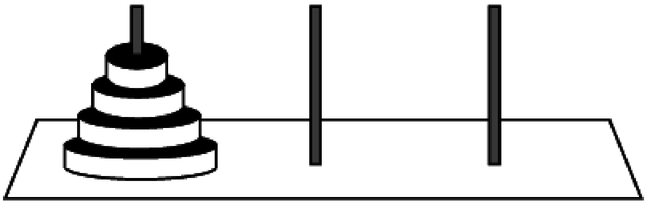


```
In [273]:
L=["S","A","T","A","N","O","S","C","I","L","L","A","T","E","M","Y","M","E","T","A","L","L","I","C","S","'
","N","A","T","A","S"]
```

Write a recursive function `def IsPalindrome(L)` witch test if a word (a list) `L` is a palindrome.

Exercise 3 – The Hanoi towers

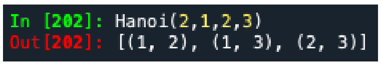
The *Hanoi towers* is a game, invented by the French mathematician Édouard Lucas. It consists of three rods, and a number of disks of different sizes which can slide on any rod. Let n be the number of disks. The game starts in the following configuration:



The scope of the game is to move all disks on the last rod, with the following rules:

1. you can move only one disk at a time,
2. each move consists of taking the upper disk from one of the stack and placing it on top of another stack,
3. no disk may be placed on top of a smaller disk.

Question 1. Write a recursive function `def hanoi(n,i,j,k)` which prints the list of moves to be made if we start from `n` disks on position `i` and the goal is to move this disks on position `k` by using the intermediate position `j`. For example,



```
In [202]: Hanoi(2,1,2,3)
Out[202]: [(1, 2), (1, 3), (2, 3)]
```

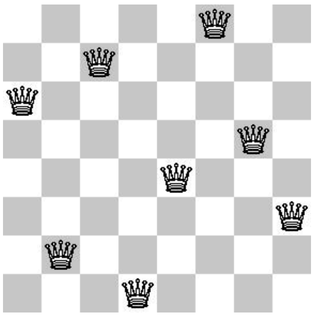
You can test your results by playing the game at :

`https://fr.goobix.com/jeux-en-ligne/tours-de-hanoi/#`

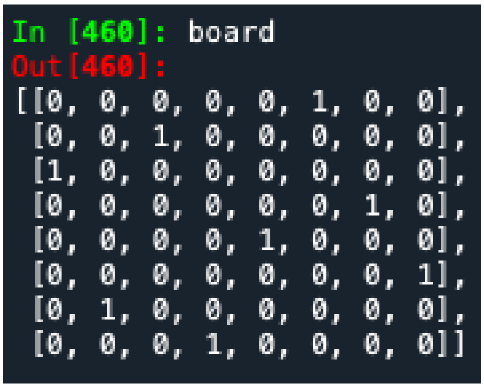
Question 2. Find out the number of moves, depending on n.

Exercise 4 – The n queens problem

We want to place n queens on an $n \times n$ chessboard, such that no queen threatens another, *i.e* we require that no two queens share the same row, column or diagonal. Here is a possible solution for $n = 8$.

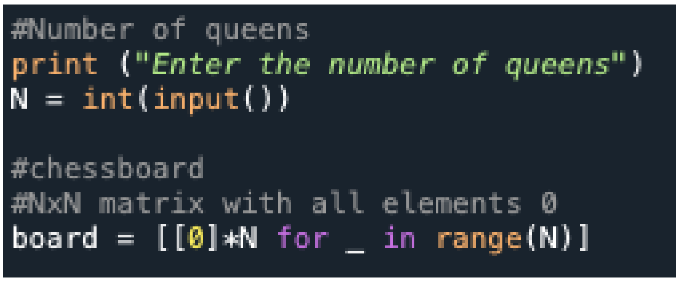


We label the rows and the columns from 0 to $n - 1$. A configuration is represented by matrix indicating the queens' positions. For example, the configuration shown in the picture is represented by



```
In [460]: board
Out[460]:
[[0, 0, 0, 0, 0, 1, 0, 0],
 [0, 0, 1, 0, 0, 0, 0, 0],
 [1, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 1, 0],
 [0, 0, 0, 0, 1, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 1],
 [0, 1, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 1, 0, 0, 0, 0]]
```

Question 1. Start by write this on the top on your script



```
#Number of queens
print ("Enter the number of queens")
N = int(input())

#chessboard
#NxN matrix with all elements 0
board = [[0]*N for _ in range(N)]
```

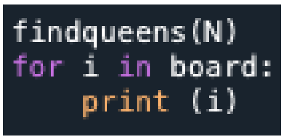
When we will run this script, the user will choice a value for `N`, the number of queens for a chessboard of dimension `NxN` and initializing the chessboard with zero queen.

Question 2. Create a function `is_attack(i,j)` which returns a boolean indicating whether the position `(i,j)` is attack by a queen in the chessboard.

Question 3. Write a function `def findqueens(n)` which returns

- a solution to the n queens problem, if it exists,
- otherwise the 0 matrix.

Question 4. End your script by



```
findqueens(N)
for i in board:
    print (i)
```