# HOMEWORK 1 - Boundaries

**Zadání :**

* hranice krajiny A a krajiny B jsou definovány neuspořádanými listy s hraničními body
  + countryA\_bounds = [“A”, “W”, “X”, “C”, “M”, “K”]
  + countryB\_bounds = [ “W”, “O”, “L”, “A”, “K”, “U”, “S”]
* vytvořte nový list countryAB\_bounds, do kterého uložíte hraniční body pro hypotetickou krajinu, která by vznikla spojením krajiny A a B
* hint: hraniční body, které se opakují v obou listech nebudou aktuální pro novou krajinu

**Řešení :**

countryA\_bounds = ['A', 'W', 'X', 'C', 'M', 'K']

countryB\_bounds = [ 'W', 'O', 'L', 'A', 'K', 'U', 'S']

countryAB\_bounds = []

# first possible solution - appending to new list

for boundA in countryA\_bounds:

if boundA not in countryB\_bounds:

countryAB\_bounds.append(boundA)

for boundB in countryB\_bounds:

if boundB not in countryA\_bounds:

countryAB\_bounds.append(boundB)

print(countryAB\_bounds)

# second possible solution - creating list with bounds to be removed

bounds\_to\_remove = []

countryAB\_bounds = countryA\_bounds + countryB\_bounds

for boundA in countryA\_bounds:

if boundA in countryB\_bounds:

bounds\_to\_remove.append(boundA)

# needs to remove twice

for bound\_remove in bounds\_to\_remove:

countryAB\_bounds.remove(bound\_remove)

countryAB\_bounds.remove(bound\_remove)

print(countryAB\_bounds)

# HOMEWORK 2 - Country Statistics

**Zadání :**

* máme list s názvy krajin countries = [“Greece”, “Czech Republic”, “Portugal“, “Hungary”, “Sweden”], list s počtem obyvatel v 1000 population = [10769, 10535, 10311, 9835, 9794] a průměrný roční růst (v %) growth = [-0.11, -0.06, -0.56, -0.28, 1.03]
* definujte list population\_estimation, kde pro každou krajinu vypočtete odhadovaný počet obyvatel o 100 let
* můžete počítat se stejným přírůstkem/úbytkem pro celé období 100 let (není potřeba 100letý součet počítat opakovaně vždy po jednom roce, ale hodnotu z prvního roku stačí vynásobit 100)
* předpokládáme, že hodnoty pro jednotlivé krajiny jsou v rámci listů vždy na stejném indexu (Greece - index 0, Czech Republic - index 1)

**Řešení :**

countries = ['Greece', 'Czech Republic', 'Portugal', 'Hungary', 'Sweden']

population = [10769, 10535, 10311, 9835, 9794]

growth = [-0.11, -0.06, -0.56, -0.28, 1.03]

population\_estimation = []

for c in range(len(countries)):

# estimation calculation

c\_estimation = (100 \* growth[c]/100 \* population[c]) + population[c]

population\_estimation.append(c\_estimation)

print(countries[c], c\_estimation)

print(population\_estimation)

# HOMEWORK 3 - Minimum Bounding Rectangle

**Zadání :**

* definujte minimální obdélník, jehož strany jsou rovnoběžné s osami souřadnicového systému => je definovaný 4 body, které leží na souřadnicích maximálních nebo minimálních hodnot zeměpisné šířky a délky daného útvaru
* hranice Kambodže jsou definované pomocí těchto souřadnic
  + [[103.49728,10.632555], [103.09069,11.153661], [102.584932,12.186595], [102.348099,13.394247], [102.988422,14.225721], [104.281418,14.416743], [105.218777,14.273212], [106.043946,13.881091], [106.496373,14.570584], [107.382727,14.202441], [107.614548,13.535531], [107.491403,12.337206], [105.810524,11.567615], [106.24967,10.961812], [105.199915,10.88931], [104.334335,10.486544], [103.49728,10.632555]]
* vypočítejte souřadnice pro minimum bounding rectangle polygonu Kambodže
* hint: pro prohledání souřadnic použijte for cyklus a hledejte v listu maximální a minimální hodnoty

**Řešení :**

bounds = [[103.49728,10.632555], [103.09069,11.153661], [102.584932,12.186595], [102.348099,13.394247], [102.988422,14.225721], [104.281418,14.416743], [105.218777,14.273212], [106.043946,13.881091], [106.496373,14.570584], [107.382727,14.202441], [107.614548,13.535531], [107.491403,12.337206], [105.810524,11.567615], [106.24967,10.961812], [105.199915,10.88931], [104.334335,10.486544], [103.49728,10.632555]]

mbr = [[],[],[],[]]

# predefining maximal and minimal coordinate values

north = -90 # because -90 is minimal latitude value

south = 90

west = 180

east = -180

for bound in bounds:

lon = bound[0]

lat = bound[1]

# checking if value of lon or lat is minimal/maximal

if lat > north:

north = lat

elif lat < south:

south = lat

if lon > east:

east = lon

elif lon < west:

west = lon

# [northwest, northeast, southwest, southeast]

mbr = [[west, north],[east, north],[west, south],[east, south]]

print(mbr)