from bs4 import BeautifulSoup  
import ctypes  
from datetime import date  
import filecmp  
import os  
import pandas as pd  
import requests  
import re  
  
  
def exam\_region(source\_code, region):  
 # Convert source code to string  
 source\_code\_str = str(source\_code)  
  
 # Type of information  
 name = ['Concurso']  
 vagas = ['Vagas']  
 nivel = ['Nível']  
 salario = ['Salário Até']  
 inscricao = ['Inscrição Até']  
 link = ['Link']  
 combinacao\_concursos = []  
  
 # Select the region  
 if region == 'nacional':  
 initial\_tag = source\_code\_str.find('<h2>NACIONAL</h2>') + len('<h2>NACIONAL</h2>')  
 final\_tag = source\_code\_str.find('<h2>REGIÃO SUDESTE</h2>') + len('<h2>REGIÃO SUDESTE</h2>')  
  
 elif region == 'CE':  
 initial\_tag = source\_code\_str.find('<div class="uf">CEARÁ</div>') + len('<div class="uf">CEARÁ</div>')  
 final\_tag = source\_code\_str.find('<div class="uf">MARANHÃO</div>') + len('<div class="uf">MARANHÃO</div>')  
  
 elif region == 'SP':  
 initial\_tag = source\_code\_str.find('<div class="uf">SÃO PAULO</div>') + len('<div class="uf">SÃO PAULO</div>')  
 final\_tag = source\_code\_str.find('<div class="uf">RIO DE JANEIRO</div>') + len('<div class="uf">RIO DE JANEIRO</div>')  
  
 elif region == 'RJ':  
 initial\_tag = source\_code\_str.find('<div class="uf">RIO DE JANEIRO</div>') + len('<div class="uf">RIO DE JANEIRO</div>')  
 final\_tag = source\_code\_str.find('<div class="uf">MINAS GERAIS</div>') + len('<div class="uf">MINAS GERAIS</div>')  
  
 elif region == 'MG':  
 initial\_tag = source\_code\_str.find('<div class="uf">MINAS GERAIS</div>') + len('<div class="uf">MINAS GERAIS</div>')  
 final\_tag = source\_code\_str.find('<div class="uf">ESPÍRITO SANTO</div>') + len('<div class="uf">ESPÍRITO SANTO</div>')  
  
 elif region == 'ES':  
 initial\_tag = source\_code\_str.find('<div class="uf">ESPÍRITO SANTO</div>') + len('<div class="uf">ESPÍRITO SANTO</div>')  
 final\_tag = source\_code\_str.find('<h2>REGIÃO SUL</h2>') + len('<h2>REGIÃO SUL</h2>')  
  
 elif region == 'PR':  
 initial\_tag = source\_code\_str.find('<div class="uf">PARANÁ</div>') + len('<div class="uf">PARANÁ</div>')  
 final\_tag = source\_code\_str.find('<div class="uf">RIO GRANDE DO SUL</div>') + len('<div class="uf">RIO GRANDE DO SUL</div>')  
  
 elif region == 'SC':  
 initial\_tag = source\_code\_str.find('<div class="uf">SANTA CATARINA</div>') + len('<div class="uf">SANTA CATARINA</div>')  
 final\_tag = source\_code\_str.find('<h2>REGIÃO CENTRO-OESTE</h2>') + len('<h2>REGIÃO CENTRO-OESTE</h2>')  
  
 elif region == 'DF':  
 initial\_tag = source\_code\_str.find('<div class="uf">DISTRITO FEDERAL</div>') + len('<div class="uf">DISTRITO FEDERAL</div>')  
 final\_tag = source\_code\_str.find('<div class="uf">GOIÁS</div>') + len('<div class="uf">GOIÁS</div>')  
  
 elif region == 'GO':  
 initial\_tag = source\_code\_str.find('<div class="uf">GOIÁS</div>') + len('<div class="uf">GOIÁS</div>')  
 final\_tag = source\_code\_str.find('<div class="uf">MATO GROSSO DO SUL</div>') + len('<div class="uf">MATO GROSSO DO SUL</div>')  
  
 elif region == 'MS':  
 initial\_tag = source\_code\_str.find('<div class="uf">MATO GROSSO DO SUL</div>') + len('<div class="uf">MATO GROSSO DO SUL</div>')  
 final\_tag = source\_code\_str.find('<div class="uf">MATO GROSSO</div>') + len('<div class="uf">MATO GROSSO</div>')  
  
 elif region == 'MT':  
 initial\_tag = source\_code\_str.find('<div class="uf">MATO GROSSO</div>') + len('<div class="uf">MATO GROSSO</div>')  
 final\_tag = source\_code\_str.find('<h2>REGIÃO NORTE</h2>') + len('<h2>REGIÃO NORTE</h2>')  
  
 elif region == 'AM':  
 initial\_tag = source\_code\_str.find('<div class="uf">AMAZONAS</div>') + len('<div class="uf">AMAZONAS</div>')  
 final\_tag = source\_code\_str.find('<div class="uf">ACRE</div>') + len('<div class="uf">ACRE</div>')  
  
 elif region == 'AC':  
 initial\_tag = source\_code\_str.find('<div class="uf">ACRE</div>') + len('<div class="uf">ACRE</div>')  
 final\_tag = source\_code\_str.find('<div class="uf">PARÁ</div>') + len('<div class="uf">PARÁ</div>')  
  
 elif region == 'PA':  
 initial\_tag = source\_code\_str.find('<div class="uf">PARÁ</div>') + len('<div class="uf">PARÁ</div>')  
 final\_tag = source\_code\_str.find('<div class="uf">RONDÔNIA</div>') + len('<div class="uf">RONDÔNIA</div>')  
  
 elif region == 'RO':  
 initial\_tag = source\_code\_str.find('<div class="uf">RONDÔNIA</div>') + len('<div class="uf">RONDÔNIA</div>')  
 final\_tag = source\_code\_str.find('<div class="uf">TOCANTINS</div>') + len('<div class="uf">TOCANTINS</div>')  
  
 elif region == 'TO':  
 initial\_tag = source\_code\_str.find('<div class="uf">TOCANTINS</div>') + len('<div class="uf">TOCANTINS</div>')  
 final\_tag = source\_code\_str.find('<h2>REGIÃO NORDESTE</h2>') + len('<h2>REGIÃO NORDESTE</h2>')  
  
 elif region == 'AL':  
 initial\_tag = source\_code\_str.find('<div class="uf">ALAGOAS</div>') + len('<div class="uf">ALAGOAS</div>')  
 final\_tag = source\_code\_str.find('<div class="uf">BAHIA</div>') + len('<div class="uf">BAHIA</div>')  
  
 elif region == 'BA':  
 initial\_tag = source\_code\_str.find('<div class="uf">BAHIA</div>') + len('<div class="uf">BAHIA</div>')  
 final\_tag = source\_code\_str.find('<div class="uf">CEARÁ</div>') + len('<div class="uf">CEARÁ</div>')  
  
 elif region == 'MA':  
 initial\_tag = source\_code\_str.find('<div class="uf">MARANHÃO</div>') + len('<div class="uf">MARANHÃO</div>')  
 final\_tag = source\_code\_str.find('<div class="uf">PARAÍBA</div>') + len('<div class="uf">PARAÍBA</div>')  
  
 elif region == 'PA':  
 initial\_tag = source\_code\_str.find('<div class="uf">PARAÍBA</div>') + len('<div class="uf">PARAÍBA</div>')  
 final\_tag = source\_code\_str.find('<div class="uf">PERNAMBUCO</div>') + len('<div class="uf">PERNAMBUCO</div>')  
  
 elif region == 'PE':  
 initial\_tag = source\_code\_str.find('<div class="uf">PERNAMBUCO</div>') + len('<div class="uf">PERNAMBUCO</div>')  
 final\_tag = source\_code\_str.find('<div class="uf">PIAUÍ</div>') + len('<div class="uf">PIAUÍ</div>')  
  
 elif region == 'PI':  
 initial\_tag = source\_code\_str.find('<div class="uf">PIAUÍ</div>') + len('<div class="uf">PIAUÍ</div>')  
 final\_tag = source\_code\_str.find('<div class="uf">RIO GRANDE DO NORTE</div>') + len('<div class="uf">RIO GRANDE DO NORTE</div>')  
  
 elif region == 'RN':  
 initial\_tag = source\_code\_str.find('<div class="uf">RIO GRANDE DO NORTE</div>') + len('<div class="uf">RIO GRANDE DO NORTE</div>')  
 final\_tag = source\_code\_str.find('<div class="uf">SERGIPE</div>') + len('<div class="uf">SERGIPE</div>')  
  
 elif region == 'SE':  
 initial\_tag = source\_code\_str.find('<div class="uf">SERGIPE</div>') + len('<div class="uf">SERGIPE</div>')  
 final\_tag = source\_code\_str.find('<p style="text-align:center; margin:0; padding:10px 0 0 0; font-weight:bold; color:#205c98;">VISITE PERIODICAMENTE - ATUALIZAÇÃO DIÁRIA!!!</p>') + len('<p style="text-align:center; margin:0; padding:10px 0 0 0; font-weight:bold; color:#205c98;">VISITE PERIODICAMENTE - ATUALIZAÇÃO DIÁRIA!!!</p>')  
  
  
 # Web-scraping  
 concursos\_tag = source\_code\_str[initial\_tag:final\_tag]  
 concursos\_tag = BeautifulSoup(concursos\_tag, "html.parser")  
  
 for line in concursos\_tag.findAll(class\_='ca'):  
 name.append(line.find('a').text.strip()) # Institution's name  
 link.append(line.find('a', href=True)['href']) # Link  
 vagas.append(''.join(re.findall('(\d\*) vaga', str(line.find(class\_='cd'))))) # Jobs  
 nivel.append('/'.join(re.findall('Superior|Médio', str(line.find(class\_='cd'))))) # Education  
 salario.append(''.join(re.findall('R\$ \*\d\*\.\*\d\*\,\*\d\*', str(line.find(class\_='cd'))))) # Salary  
 inscricao.append(''.join(re.findall('\d+/\d+/\d+', str(line.find(class\_='ce'))))) # Subscription date  
  
 # Merge lists  
 combinacao\_concursos.extend([list(i) for i in zip(name, vagas, nivel, salario, inscricao, link)])  
  
 return combinacao\_concursos  
  
  
  
def new\_exam():  
 novos\_concursos = ['Concursos novos disponíveis: ']  
  
 # Check if there is an old 'ConcursosAtivos.csv' file  
 if os.path.isfile('ConcursosAtivos.csv') is False:  
 os.rename('ConcursosAtivos' + date\_now + '.csv', 'ConcursosAtivos.csv')  
  
 else:  
 if filecmp.cmp('ConcursosAtivos.csv', 'ConcursosAtivos' + date\_now + '.csv') is False:  
 antigo = pd.read\_csv('ConcursosAtivos.csv', encoding='utf-16', header=None, sep = "\t")  
 novo = pd.read\_csv('ConcursosAtivos' + date\_now + '.csv', encoding='utf-16', header=None, sep = "\t")  
  
 # Find the new exam name  
 for contador in range(1, novo.shape[0]):  
 encontrou = 0  
 for contador2 in range(1, antigo.shape[0]):  
 if novo.iloc[contador, 0] == antigo.iloc[contador2, 0] and novo.iloc[contador, 1] == antigo.iloc[contador2, 1]:  
 encontrou = 1  
 if encontrou == 0:  
 print(novo.iloc[contador,0])  
 novos\_concursos.append(str(novo.iloc[contador,0])+' - '+str(novo.iloc[contador,2]))  
  
 os.remove('ConcursosAtivos.csv')  
 os.rename('ConcursosAtivos' + date\_now + '.csv', 'ConcursosAtivos.csv')  
  
 if len(novos\_concursos) > 1:  
 ctypes.windll.user32.MessageBoxW(0, '\n'.join(novos\_concursos), "Novo Concurso", 1)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 # Date  
 today = date.today()  
 date\_now = today.strftime("%d%m%Y")  
  
 # Get source code  
 LINK = "https://www.pciconcursos.com.br/concursos/"  
 response = requests.get(LINK)  
 soup = BeautifulSoup(response.text, "html.parser")  
  
 # Extract multiple states  
 # state1 = exam\_region(soup, 'nacional')  
 # state2 = exam\_region(soup, 'PR')  
 # state = state1 + state2[1:]  
  
 # Extract one state  
 state = exam\_region(soup, 'nacional')  
  
 # Save as CSV  
 df = pd.DataFrame(state)  
 df = df.replace(r'^\s\*$', '-', regex=True)  
 with open('ConcursosAtivos' + date\_now + '.csv', 'a', encoding='utf-16', newline='') as f:  
 df.to\_csv(f, encoding = 'utf-16', header = False, sep = "\t", index = False)  
  
 # Check for new exam  
 new\_exam()

Aeronáutica  
Banco do Brasil  
Marinha do Brasil  
Marinha do Brasil  
TST - Tribunal Superior do Trabalho  
Marinha do Brasil  
Marinha do Brasil  
Marinha do Brasil  
PETROBRAS - Petróleo Brasileiro S.A.