The Project

1. This is a project with minimal scaffolding. Expect to use the the discussion forums to gain insights! It's not cheating to ask others for opinions or perspectives!
2. Be inquisitive, try out new things.
3. Use the previous modules for insights into how to complete the functions! You'll have to combine Pillow, OpenCV, and Pytesseract
4. There are hints provided in Coursera, feel free to explore the hints if needed. Each hint provide progressively more details on how to solve the issue. This project is intended to be comprehensive and difficult if you do it without the hints.

The Assignment

Take a ZIP file (https://en.wikipedia.org/wiki/Zip_(file_format)) of images and process them, using a library built into python (https://docs.python.org/3/library/zipfile.html) that you need to learn how to use. A ZIP file takes several different files and compresses them, thus saving space, into one single file. The files in the ZIP file we provide are newspaper images (like you saw in week 3). Your task is to write python code which allows one to search through the images looking for the occurrences of keywords and faces. E.g. if you search for "pizza" it will return a contact sheet of all of the faces which were located on the newspaper page which mentions "pizza". This will test your ability to learn a new (library (https://docs.python.org/3/library/zipfile.html)), your ability to use OpenCV to detect faces, your ability to use tesseract to do optical character recognition, and your ability to use PIL to composite images together into contact sheets.

Each page of the newspapers is saved as a single PNG image in a file called images.zip (./readonly/images.zip). These newspapers are in english, and contain a variety of stories, advertisements and images. Note: This file is fairly large (~200 MB) and may take some time to work with, I would encourage you to use small_img.zip (./readonly/small_img.zip) for testing.

Here's an example of the output expected. Using the small_img.zip (./readonly/small_img.zip) file, if I search for the string "Christopher" I should see the following image:

Christopher Search
If I were to use the images.zip (./readonly/images.zip) file and search for "Mark" I should see the following image (note that there are times when there are no faces on a page, but a word is found!):

Mark Search

Note: That big file can take some time to process - for me it took nearly ten minutes! Use the small one for testing.

In [1]:
```
# Python 3 Specialization Final Project
# Edited to run in Coursera Jupyter Notebook to use the files in the server.

import cv2 as cv
import numpy
import PIL
from PIL import Image, ImageDraw, ImageFont
from PIL import ImageEnhance
import pytesseract
import zipfile

#########################################################################
##### FUNCTIONS ###########################################################
#########################################################################

# Function that takes an image taken from Image.open(file) and resizes it:
def image_resize_pil(file_image_pil,int_perc_resize):
    w,h = file_image_pil.size
    w_res = int(w*int_perc_resize/100)
    h_res = int(h*int_perc_resize/100)
    img_resize_new_pil = file_image_pil.resize((w_res, h_res))
    return img_resize_new_pil

# Function that takes an image taken from cv.imread(file) and resizes it:
def image_resize_cv(file_image_cv,int_perc_resize):
    width = int(file_image_cv.shape[1] * int_perc_resize / 100)
    height = int(file_image_cv.shape[0] * int_perc_resize / 100)
    dim = (width, height)
    # resize image
    img_resize_new_cv = cv.resize(file_image_cv, dim, interpolation = cv.INTER_ARE
A)
    return img_resize_new_cv



# Function that takes an image taken from Image.open(file),
# and extracts the words using pytesseract,
# in the form of a list of words
def image_get_words(file_image_pil):
    # pytesseract.pytesseract.tesseract_cmd = r"C:\Program Files (x86)\Tesseract-OC
R\tesseract.exe"
    # print(pytesseract.image_to_string(image))
    print("Word list generation in progress. Please be patient...")
    str_words = pytesseract.image_to_string(file_image_pil)
    str_words = str_words.strip()
    str_words = str_words.replace("\n", " ").replace("\t", " ")
    str_words = str_words.replace(".", " ").replace(",", " ").replace(":", " ").rep
lace(";", " ")
    lst_words = str_words.split(" ")
    lst_words_unique = []
    for word in lst_words:
        if word not in lst_words_unique:
            lst_words_unique.append(word)

    return lst_words_unique



# Function that takes an image taken from Image.open(file),
# scaleFactor, min width, and min height
# and extracts the faces found using cv,
# which must be larger than the min width and min height
# in the form of a list of images
def image_get_faces(file_image_pil, scaleFactor, min_w=100, min_h=100):
```

```
Word to search: Mark
Scalefactor to use in face recognition (typical 1.01-1.5, 1.35 recommended): 1.3
5
Initializing analysis of readonly/images.zip
Word to search: Mark
Face scalefactor used: 1.35
Analyzing file: a-0.png in readonly/images.zip
File 1 of 14
Word list generation in progress. Please be patient...
Word match is: True
Face ID in progress. Please be patient...
Number of faces found: 6
File a-0.png analysis complete!
Analyzing file: a-1.png in readonly/images.zip
File 2 of 14
Word list generation in progress. Please be patient...
Word match is: True
Face ID in progress. Please be patient...
Number of faces found: 5
File a-1.png analysis complete!
Analyzing file: a-10.png in readonly/images.zip
File 3 of 14
Word list generation in progress. Please be patient...
Word match is: False
Number of faces found: 0
No faces found
File a-10.png analysis complete!
Analyzing file: a-11.png in readonly/images.zip
File 4 of 14
Word list generation in progress. Please be patient...
Word match is: False
Number of faces found: 0
No faces found
File a-11.png analysis complete!
Analyzing file: a-12.png in readonly/images.zip
File 5 of 14
Word list generation in progress. Please be patient...
Word match is: False
Number of faces found: 0
No faces found
File a-12.png analysis complete!
Analyzing file: a-13.png in readonly/images.zip
File 6 of 14
Word list generation in progress. Please be patient...
Word match is: False
Number of faces found: 0
No faces found
File a-13.png analysis complete!
Analyzing file: a-2.png in readonly/images.zip
File 7 of 14
Word list generation in progress. Please be patient...
Word match is: True
Face ID in progress. Please be patient...
Number of faces found: 3
File a-2.png analysis complete!
Analyzing file: a-3.png in readonly/images.zip
File 8 of 14
Word list generation in progress. Please be patient...
Word match is: True
Face ID in progress. Please be patient...
Number of faces found: 1
File a-3.png analysis complete!
Analyzing file: a-4.png in readonly/images.zip
File 9 of 14
```

Results found in a-0.png

Final Project by:
Guillermo H. Kumazawa

Results found in a-1.png

No results found in a-10.png

No results found in a-11.png

No results found in a-12.png

No results found in a-13.png

Results found in a-2.png

Results found in a-3.png

No results found in a-4.png

```
All done!
```

In [ ]: