# The Project

Adam Herrman

Take a .zip file of images and process them, using python libraries to search for words and faces. The files in the ZIP file are .png images of archived newspapers. These newspapers are in english, and contain a variety of stories, advertisements and images. The task is to allow one to search through the images looking for the occurrences of keywords and faces. E.g. if you search for "pizza" the program will return a contact sheet of all of the faces which were located on a newspaper page which mentions "pizza". This project uses the libraries: OpenCV to detect faces, tesseract to do optical character recognition, and Pillow to composite images together into contact sheets.

## Import Libraries

In [1]:
```
import zipfile
import PIL
from PIL import Image
from PIL import ImageDraw
import pytesseract
import cv2 as cv
import numpy as np

# loading the face detection classifier
face_cascade = cv.CascadeClassifier('readonly/haarcascade_frontalface_default.
xml')
```

## Define a function to extract images from zip files and open them.

In [2]:
```
def unzip(file_str):
    filenames = []
    images_dictionary = {}
    z = zipfile.ZipFile(file_str, "r")

    for item in z.infolist():
        file  = z.open(item.filename)
        image = Image.open(file)
        images_dictionary[item.filename] = image
    return images_dictionary
```

## Define a function to reconize words and search for key words in the image.

In [3]:
```
def word_search(word_str, images_dictionary):
    search_results = []
    for image_name in images_dictionary:
        if word_str in pytesseract.image_to_string(images_dictionary[image_nam
e]):
            search_results.append(image_name)
    return search_results
```

## Define a fuction to search for faces within images and record them to a contact sheet.

In [4]:
```
def face_search(images_dictionary, search_results):
    for result in search_results:
        print("Results found in file {}".format(result))
        face_list = []

        try:
            faces = (face_cascade.detectMultiScale(np.array(images_dictionary[
result]),1.35,4)).tolist()
            for x, y, w, h in faces:
                face = images_dictionary[result].crop((x,y,x+w,y+h))
                face_list.append(face)
            contact_sheet = PIL.Image.new(images_dictionary[result].mode, (500
,100+(len(face_list)//5)*100))
            x = 0
            y = 0

            for face in face_list:
                face.thumbnail((100, 100))
                contact_sheet.paste(face, (x, y))
                if x == 400:
                    x = 0
                    y = y + 100
                else:
                    x = x + 100
            display(contact_sheet)

        except:
            print("But there were no faces in the file!")
```

## Define a function to put the previous 3 functions together.

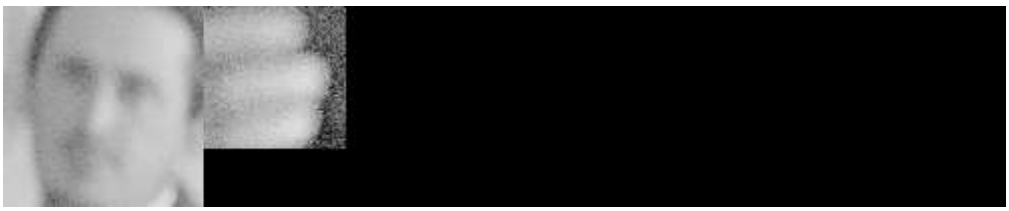
In [5]:
```
def search(word_str, filepath):
    images_dictionary = unzip(filepath)
    search_results = word_search(word_str, images_dictionary)
    face_search(images_dictionary, search_results)
```

## Search for faces in images containing the name Christopher

In [6]:
```
search("Christopher","readonly/small_img.zip")
```

```
Results found in file a-0.png
```

```
Results found in file a-3.png
```



## Search for faces in images containing the name Mark

In [7]:
```
search("Mark","readonly/images.zip")
```

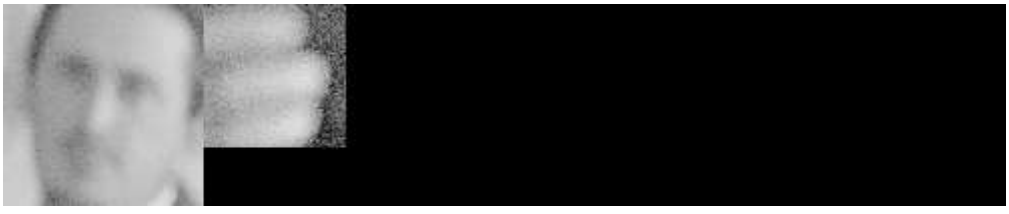
```
Results found in file a-0.png
```

```
Results found in file a-1.png
```

```
Results found in file a-10.png
But there were no faces in the file!
Results found in file a-13.png
```

```
Results found in file a-2.png
```

```
Results found in file a-3.png
```



```
Results found in file a-8.png
But there were no faces in the file!
```

In [ ]: