In [1]:

```
import cv2
import numpy as np
```

In [2]:

```
%matplotlib inline
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
```

In [3]:

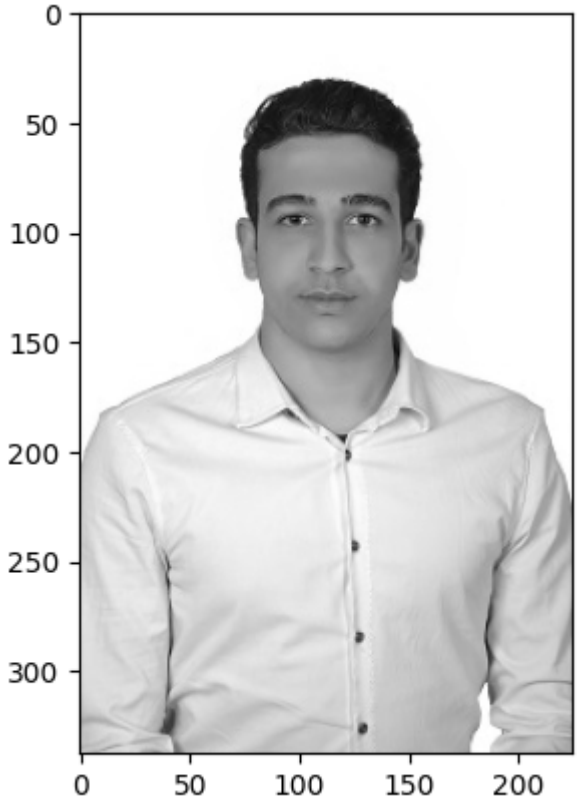
```
def show(img):
    if (len(img.shape) == 2):
        plt.imshow(sml, cmap='gray')
    else:
        plt.imshow(big)
```

In [4]:

```
big = mpimg.imread('big.jpeg')
show(big)
```

In [5]:

```
sml = cv2.imread('sml.png')
sml = cv2.cvtColor(sml, cv2.COLOR_RGB2GRAY)
show(sml)
```



In [6]:

```
sml_w, sml_h = sml.shape
big_w, big_h,_ = big.shape
```

In [7]:

```
def img2fv(img):
    if (len(img.shape) == 2):
        return img.reshape(img.shape[0] * img.shape[1])
    else:
        return img.reshape(img.shape[0] * img.shape[1] * img.shape[2])
```

In [8]:

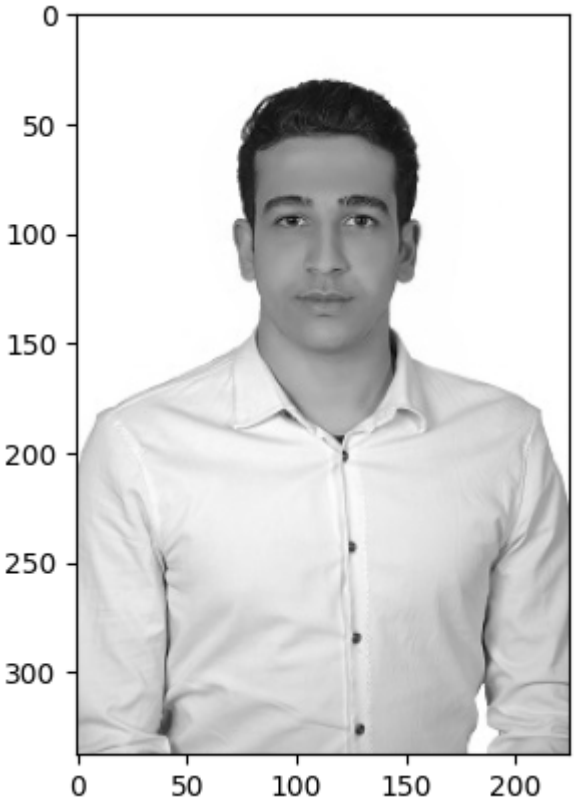
```
def fv2img(fv,h, w, gray = False):
    if (gray):
        return np.reshape(fv, (w, h))
    else:
        return np.reshape(fv, (w, h, 3))
```

In [9]:

```
#recover test
show(fv2img(img2fv(big), big_h, big_w))
```

In [10]:

```
#recover test
show(fv2img(img2fv(sml), sml_h, sml_w, True))
```



In [11]:

```
sml_fv = img2fv(sml)
big_fv = img2fv(big)
result_fv = np.empty(len(big_fv), dtype=int)
```

In [12]:

```
print((big_w*big_h*3) / (sml_w * sml_h) > 8)
```

```
True
```

In [13]:

```
def hide_num(tar, offset):
    for i in range(8):
        shifted = tar >> i
        index = offset + i
        if (shifted % 2 == 0):
            if (big_fv[index] % 2 != 0):
                result_fv[index] = big_fv[index] + 1
            else:
                result_fv[index] = big_fv[index]
```

In [14]:

```
offset = 0
for tar in sml_fv:
    hide_num(tar, offset)
    offset += 8
```

In [15]:

```
show(fv2img(result_fv, big_h, big_w))
```

In [16]:

```
recovered_fv = np.empty(len(sml_fv), dtype=int)
```

In [18]:

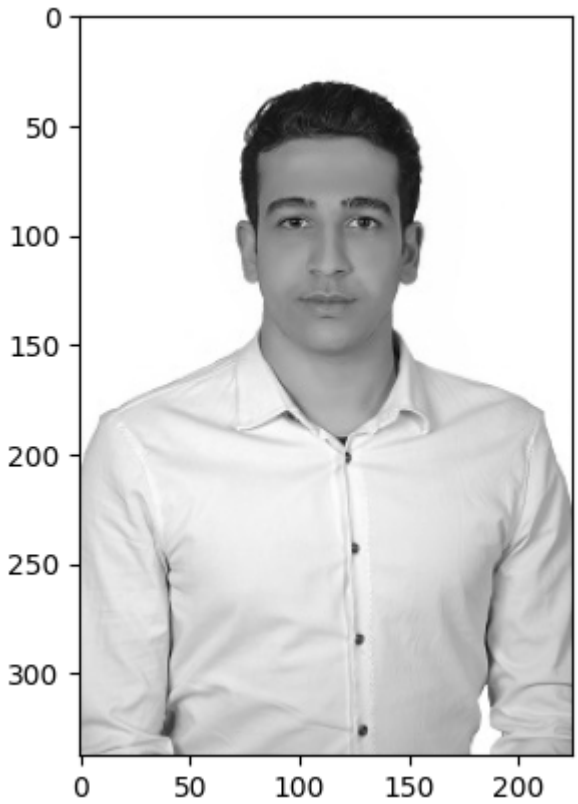
```
def recover_num(tar, offset):
    num = 0
    for i in range(8):
        index = offset + i
        if (result_fv[index] % 2 != 0):
            num += pow(2, i)
    return num
```

In [19]:

```
offset = 0
for i, tar in enumerate(sml_fv):
    recovered_fv[i] = recover_num(tar, offset)
    offset += 8
```

In [20]:

```
show(fv2img(recovered_fv, sml_h, sml_w, True))
```



In [ ]:

```

```