```
# OpenCV program to detect face in real time
# import libraries of python OpenCV
# where its functionality resides
import cv2
import math
import numpy as np
# capture frames from a camera
cap = cv2.VideoCapture(0)

# loop runs if capturing has been initialized.
while 1:

    # reads frames from a camera
    ret, img = cap.read()
    # convert to gray scale of each frames
    rows,cols,p = img.shape
    sizez = rows,cols,p
    size = rows,cols
    if rows>cols:
        pn = cols
    else :
        pn = rows
    y1 = np.zeros(size, dtype=bool) # ?
    # making logarithmic spiral on image where r = a*exp(b*q)
            #where q is phase angle and r is radius
    # the radius of the spiral defined become twice on every rotation
    # so,  a*exp(b*(q+2*pi) = 2* a*(exp(b*q))
    #   => b = log(2)/2*pi
    #  largest qm has been defined such as all point (r,q) resides in image
    m = pn/10
    pp=math.pi
    mm=math.log2(m)
    qm=mm*2*pp # qm is the largest value of the spiral where it ends starting from
centre
    q=qm
    b = math.log(2)/(2*pp) # b has been calculated above
    a = m/(math.exp(2* b*pp)*mm) # a is calculated from the main equation putting
(m,qm) on it as r is m when q=qm
    k = int(pn/2)   # radius range can be from 0 to k
    for rm in range(k,int(k/2),-1):  # for the same angle qm, radius will be k and
k/2 after 2*pi
                                        # to fill the gap we take spiral of all the
radius ranging from k/2 to k
        q=qm
        for r in range(int(3*rm),int(1*rm),-1): #loop followed by change in r
            dq = (math.log(r/(r+1)))/b     # for some minute change in r dq(rate of
q change) is calculated
            q = q+dq
```

```
            r1 = r/3
            x = k + r1*math.cos(q)
            y = k + r1*math.sin(q)
            if img[int(x),int(y),2] <  21 or img[int(x),int(y),1] <  21 or
img[int(x),int(y),0] <  19 or y1.item(int(x),int(y)) == 1 :
                        continue
            y1.itemset((int(x),int(y)) ,1)
            img[int(x),int(y),2] = img[int(x),int(y),2]+50
            #if img[int(x),int(y),2]  1:
             #   img[int(x),int(y),2]= 1

    # Detects faces of different sizes in the input image
    cv2.imshow('img',img)

    # Wait for Esc key to stop
    k = cv2.waitKey(30) & 0xff
    if k == 27:
        break

# Close the window
cap.release()

# De-allocate any associated memory usage
cv2.destroyAllWindows()
```