In [6]:

```
import cv2
import numpy as np

frameWidth = 640
frameHeight = 480

vcap = cv2.VideoCapture(0)
vcap.set(3,frameWidth)
vcap.set(4,frameHeight)
vcap.set(10,150)

myColors = [[5,107,0,19,255,255],   # Hue Min, Saturation Min, Value Min, Hue Max, Saturati
            [133,56,0,159,156,255],
            [57,76,0,100,255,255],
            [90,48,0,118,255,255]]

myColorValues = [[51,153,255],   #Orange           ## BGR
                 [255,0,255],    #Purple
                 [0,255,0],      #Green
                 [255,0,0]]      #Blue

myPoints = []  ##  [x,y, colorId_]


def findColor(img,myColors,myColorValues):
    imgHSV = cv2.cvtColor(img,cv2.COLOR_BGR2HSV)
    count =0
    newPoints = []
    for color in myColors:
        lower = np.array(color[:3])
        upper = np.array(color[3:6])
        mask = cv2.inRange(imgHSV, lower, upper)
        x,y=getContours(mask)
        cv2.circle(imgResult,(x,y),10,myColorValues[count],cv2.FILLED)
        if x!=0 and y!=0:
            newPoints.append([x,y,count])

        count+=1
    return newPoints
def getContours(img):
    contours,hierarchy = cv2.findContours(img, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
    x,y,w,h=0,0,0,0
    for cnt in contours:
        area = cv2.contourArea(cnt)
        if area >=500:
            #cv2.drawContours(imgResult, cnt, -1, (255, 0, 0), 3)
            peri = cv2.arcLength(cnt,True)
            approx = cv2.approxPolyDP(cnt,0.02*peri,True)
            x, y, w, h = cv2.boundingRect(approx)
    return x+w//2,y

def drawOnCanvas(myPoints,myColorValues):
    for point in myPoints:
        cv2.circle(imgResult, (point[0], point[1]), 10, myColorValues[point[2]], cv2.FILLED

while True:
    success, img = vcap.read()
    imgResult = img.copy()
    newPoints=findColor(img,myColors,myColorValues)
```

```
    if len(newPoints)!=0:
        for newP in newPoints:
            myPoints.append(newP)
    if len(myPoints)!=0:
        drawOnCanvas(myPoints,myColorValues)
    cv2.imshow("Result",imgResult)
    if cv2.waitKey(1) & 0xFF ==ord('q'):
        break
```

In [ ]: