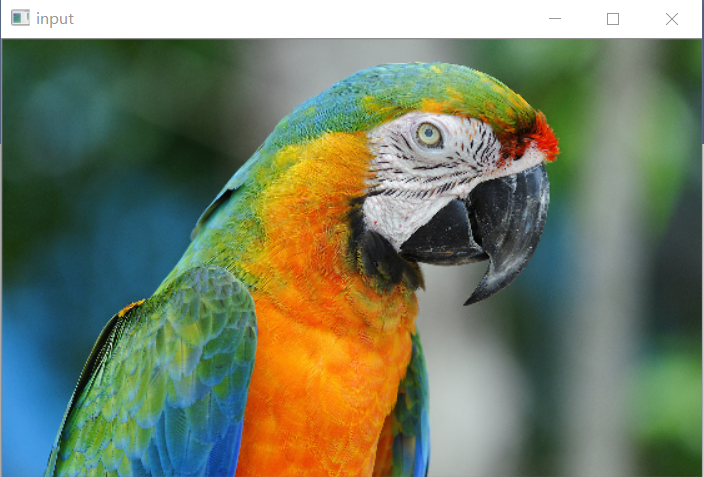
Task №1. Read the image from parrot.png and print the number of columns of this image.

**import** cv2 **as** cv  
src=cv.imread(**"D:/picture1/parrot.png"**)  
h,w=src.shape[:2]  
print(src.shape[2])  
dst=cv.resize(src,(w//3,h//3),interpolation=cv.INTER\_LINEAR)  
cv.imshow(**"input"**,dst)  
cv.waitKey(0)  
cv.destroyAllWindows()

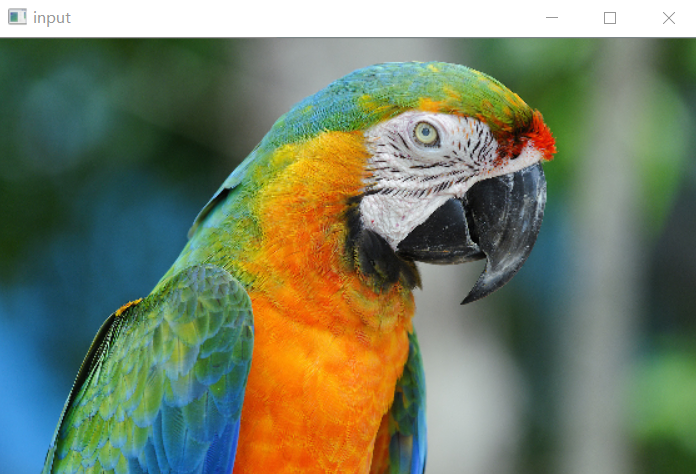
3



Task №2. Read the image from img.png. This image has an odd number of rows and columns. Change the center pixel color of this image to rgb green (102, 204, 102) and change all black pixels with white pixels. Save the image to an out\_img.png.

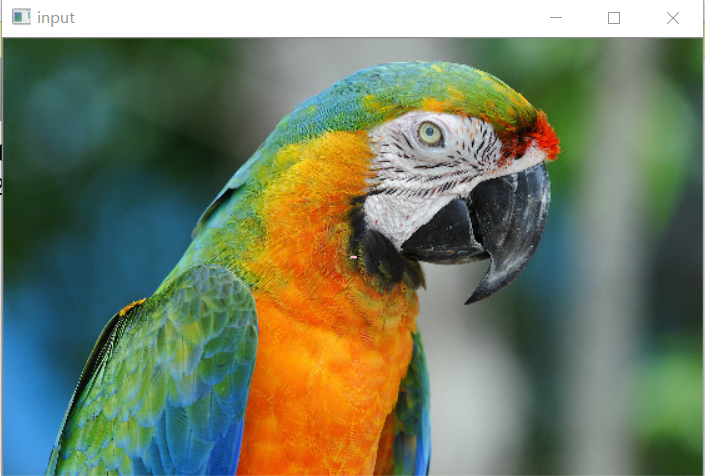
**import** cv2 **as** cv  
**import** numpy **as** np  
src=cv.imread(**"D:/picture1/parrot.png"**)  
h,w=src.shape[:2]  
print(src.shape)  
src[w//2+1,h//2+1]=[102,204,102]  
*# loc=np.where(src==[0,0,0])  
# print(loc)  
# for pt in zip(\*loc[::-1]):***for** row **in** range(h):  
 **for** col **in** range(w):  
 b,g,r=src[row,col]  
 **if** b==0 **and** g==0 **and** r==0:  
 src[row,col]=[255,255,255]  
cv.imwrite(**"D:/picture2/out\_img.png"**,src)  
dst=cv.resize(src,(w//3,h//3),interpolation=cv.INTER\_LINEAR)  
cv.imshow(**"input"**,dst)  
cv.waitKey(0)  
cv.destroyAllWindows()

(1050, 1680, 3)



Task №3. Read the image from out\_img.png. In the center of this image, draw a rectangle the size of 7 rows and 15 columns. Make it a pink rgb color (255, 192, 203) and save it to the out\_img.png.

**import** cv2 **as** cv  
**import** numpy **as** np  
src=cv.imread(**"D:/picture2/out\_img.png"**)  
h,w=src.shape[:2]  
print(src.shape)  
cv.rectangle(src,(w//2-7,h//2-3),(w//2+7,h//2+3),(203,192,255),-1,8,0)  
cv.imwrite(**"D:/picture2/out\_img.png"**,src)  
dst=cv.resize(src,(w//3,h//3),interpolation=cv.INTER\_LINEAR)  
cv.imshow(**"input"**,dst)  
cv.waitKey(0)  
cv.destroyAllWindows()



Task №4. Read the image from frame.png. The image consists of a solid color frame and the inside part of the image. The color of the frame can be found by looking at the top left pixel. The frame can have a different width on all four sides. Determine the size of the frame and print these dimensions separated by a space. Output frame sizes in the following order: left, top, right, bottom.

**import** cv2 **as** cv  
**import** numpy **as** np  
src=cv.imread(**"D:/picture1/frame.png"**)  
h,w=src.shape[:2]  
print(src.shape)  
gray=cv.cvtColor(src,cv.COLOR\_BGR2GRAY)  
ret,binary=cv.threshold(gray,0,255,cv.THRESH\_BINARY|cv.THRESH\_OTSU)  
image,contours,hierachy=cv.findContours(binary,cv.RETR\_EXTERNAL,cv.CHAIN\_APPROX\_SIMPLE)  
**for** i **in** range(len(contours)):  
 x,y,ww,hh=cv.boundingRect(contours[i])  
 print(x,y,ww,hh)  
print(**"left:"**,x)  
print(**"top:"**,y)  
print(**"right:"**,w-x-ww)  
print(**"bottom:"**,h-y-hh)  
cv.imshow(**"input"**,src)  
cv.waitKey(0)  
cv.destroyAllWindows()

(594, 879, 3)

30 30 630 530

left: 30

top: 30

right: 219

bottom: 34

