

## Flipping and Background Substitution

### 1. Flipping

```
img_w = width(img)
img_h = height(img)
mirrorPt = img_w // 2
for i in range(img_h):
    for j in range(mirrorPt):
        (rl, gl, bl) = img[i][j]
        (rr, gr, br) = img[i][img_w - j - 1]
        img[i][j] = (rr, gr, br)
        img[i][img_w-j-1] = (rl, gl, bl)
```

#### What does this code do?

```
img_w = width(img)
img_h = height(img)
magic = img_w // 2
for i in range(img_h):
    countingDown = img_w - 1
    for j in range(magic):
        (r, g, b) = img[i][j]
        img[i][countingDown] = (r, g, b)
        countingDown -= 1
```

- A. Copies top half into bottom half not mirrored.
- B. Copies left half into right half not mirrored.
- C. Mirrors around vertical axis, left into right
- D. Mirrors around horizontal axis, top into bottom
- E. Some other bizarre transformation

#### What is the size of the red box made by the following code?

```
for i in range(1, 5):
    for j in range(40, 50):
        img[i][j] = (255, 0, 0)
```

- | width                | height |
|----------------------|--------|
| A. 10                | 5      |
| B. 9                 | 4      |
| C. 5                 | 10     |
| D. 4                 | 9      |
| E. None of the above |        |

What are correct loops to make a black box of width w and height h?

```
def foo(img):  
    img_w = width(img)  
    img_h = height(img)  
    magic = img_w // 2  
    #Nested loop codes here  
    img[row][col] = (0, 0, 0)
```

A) for row in range(h+1):  
 for col in range(w+1):

B) for row in range(10, h+10):  
 for col in range(20, w+20):

C) for row in range(h):  
 for col in range(w):

D) for row in range(h, 0, -1):  
 for col in range(w, 0, -1):

E. More than one of these

## 2. Background Substitution

Nested for loop offers us a way to go through a rectangular area of a picture.

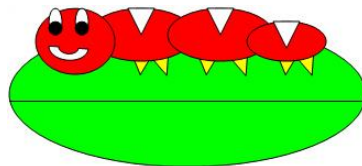
```
for i in range(start_r, end_r):  
    for j in range(start_c, end_c):  
        #do something in this region
```

But what if we want to change pixels from an irregular shape?

Remove red eye



make the caterpillar yellow



So we normally have to pick a region to go through, and check the pixel's color, and decide if we need to process this pixel.

What is the color of the caterpillar?

A. (255, 0, 0)      B. (0, 255, 0)      C. (250, 20, 20)      D. Hard to tell

Complete the code to change all the pixels' color from color to yellow.

```
def recolor(img, color, xmin, xmax, ymin, ymax):  
    for i in range(____):  
        for j in range(____):  
            (r, g, b) = img[i][j]  
            if _____:  
                img[i][j] = (255, 255, 0)
```

- If we call the function we wrote before on the caterpillar picture, what will happen?
- A. Caterpillar will turn yellow
  - B. Nothing will happen to caterpillar
  - C. Some pixels will change to yellow
  - D. You will see an error

**Key concept:** Due to different shades of colors, we need to be “fuzzy” on matching colors. So use a distance instead of exact match.

**Coding Challenges:**

Change the caterpillar to yellow. Complete a color distance function. Also use it to change the red eye of jenny. Jenny’s red eye’s rgb value is about (167, 34, 63). Think about how to call the function too.