1. For method (b) in slide 15, draw what the allocation will look like in memory.

![Memory Allocation Diagram]

2. Write the code to safely release the memory allocated by method (b).

```c
delete [] image[0];
delete [] image;
```

3. List one advantage and one disadvantage for each of the two methods on slide 21.

**3D Array:**

Easy to reach colors in a loop and r,c means the row/col in the image.

**2D Array:**

Easier to treat every pixel the same way. Decreased complexity. Easy to have varying numbers of planes.

4. Write code to allocate an RGB image that will only require two calls to “new” (as in the (b) method for monochrome images) (on the back).
unsigned char **image = new unsigned char *[height];
image[0] = new unsigned char *[height * width * 3];
for(int r=1; r<height; r++)
{
    image[r] = image[r] + r * (width * 3);
}

5. Assuming a 1000 wide by 500 tall image, what is the memory requirement (in BYTES) for each of the following image types:

a) Monochrome

1000 * 500 = 500,000

b) RGB Color

1000 * 500 * 3 = 1,500,000

c) RGBA Color

1000 * 500 * 4 = 2,000,000

d) Indexed Color

1000 * 500 * 1 + 256 * 3 = 500,768