

Test 1 will be Friday (Feb. 20) in class. It will be open notes and textbook, but no computer -- only pencil and paper. Topics on the exam include (but are not limited to):

### **Chapter 1: Intro. to Computer Science and Media Computation**

terminology - algorithm, program, media computation

computer components: CPU/processor, main memory/RAM, hard-disk

data representation: binary numbers, ASCII characters, pixel RGB

data structure: 1-D array

digitization: analog-to-digital, pixels, sound samples

The types of questions could be:

- short answer (i.e., a couple sentences) questions
- converting between binary and decimal numbers

### **Chapter 2: Intro. to Programming**

program components: comments, variables, assignment statements, reserved words, operators, functions, print

types: integers, floating-point numbers, strings

operator precedence and associativity, casting of types

defining and using functions - parameters into, returning values, local variables

JES picture functions: pickAFile, makePicture, show

JES sound functions: makeSound, play

JES shape drawing functions: addLine, addOval, addOvalFilled, addRect, addRectFilled, addArc, addArcFilled

The types of questions could be:

- predict the results of a several lines of code -- arithmetic result, variable assignment, drawing of picture
- writing several lines of code to perform a small task

### **Chapter 3: Modifying Pictures Using Loops**

picture encoding: matrix of pixels, RGB color value per pixel

for-loop with getPixels function to create picture filters

picture filters: increasing/decreasing RGB components, lighten/darken, grayscale

JES picture functions: getPixels, getPixel, getWidth, getHeight, writePictureTo

JES pixel functions: getRed, getGreen, getBlue, setRed, setGreen, setBlue, getColor, setColor, getX, getY

JES color functions: makeColor, pickAColor, makeDarker, makeLighter

The types of questions could be:

- predict the results of a filters effect on an image
- writing a filter to perform some task or complete the code of a partial filter

### **Chapter 4: Modifying Pixels in a Range**

range function

nested for-loops to iterate over a range of x, y coordinates

copying pixels

partial picture manipulations: mirroring, general copying, creating a collage, rotating 90°, scaling

JES functions: setMediaPath, getMediaPath, makeEmptyPicture

The types of questions could be:

- predict the results (changes to the picture) of a nested for-loops that iterate over a range of x, y coordinates
- writing nested for-loops to perform some task or complete a partial one

## Chapter 5: Picture Techniques with Selection and Combination

if, if-else, if-elif-else statements

Comparison operators (`==`, `<`, `>`, `<=`, `>=`, `!=`) and Boolean operators (`not`, `and`, `or`)

Usage of luminance or distance function when comparing colors for picture processing examples:

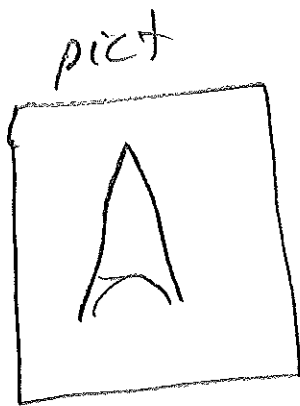
- posterization
- edge detection
- background subtraction/Chromakey

blurring by averaging pixels surrounding pixels

blending overlapping pictures by averaging each color component

The types of questions could be:

- predict the results (changes to the picture) of a nested for-loops that iterate over a range of x, y coordinates and selectively modify some pixels
- writing nested for-loops and if statements to perform some task or complete a partial one



```
main()
{
```

```
targetPict = makeCollage(pict)
```

```
def makeCollage(pict):
```

```
    collagePict = makeEmptyPicture(
```

```
        for x in range(getWidth(pict)):
```

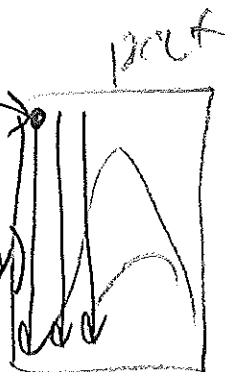
```
            for y in
```

```
                sourcePixel = getPixel(pict, x, y)
```

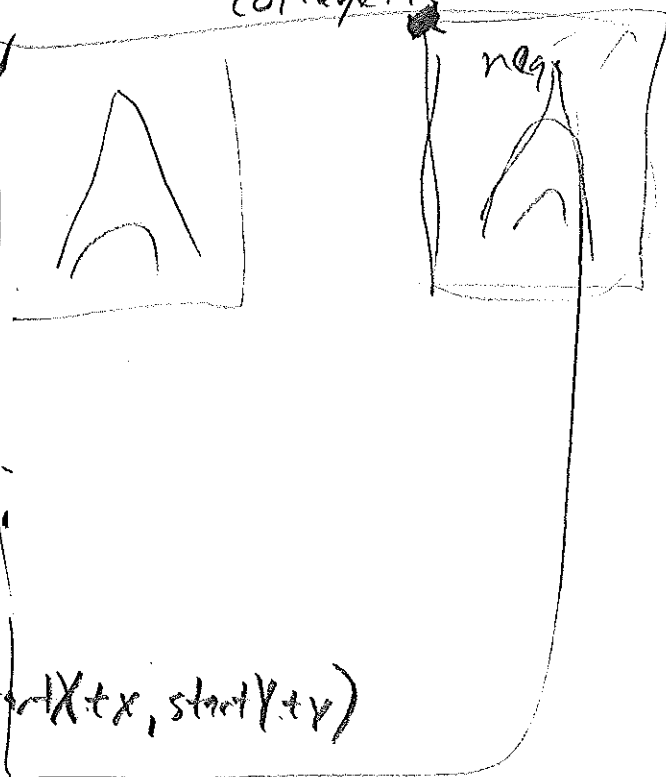
```
                collagePixel = getPixel(collagePict, x, y)
```

```
                sourceColor = getColor(sourcePixel)
```

```
                setColor(collagePixel, sourceColor)
```



collagePict



```
def makeSunset(pict, collagePict,
               startX, startY):
```

```
    for x in range(getWidth(pict)):
```

```
        for y
```

```
            sourcePixel
```

```
            collagePixel = (collagePict, startX+x, startY+y)
```