Texas International College

LAB REPORT

on

Multimedia Computing

BSc. CSIT 5th Semester

Submitted by:	**Submitted to:**
Kishor Upadhyaya	**Saroj Ghimire**
Roll no : 17	Lecturer

LAB-1
Write a program in Python to convert text to speech

PROGRAM:
```
from gtts import gTTS
import os
text_to_speak = "Hello, I am Kishor Upadhyaya"
tts = gTTS(text=text_to_speak, lang='en')
tts.save("output.mp3")
os.system("afplay output.mp3")
```

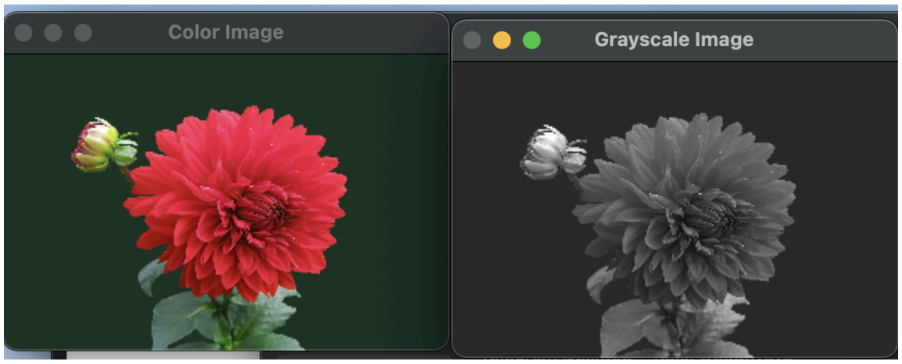
LAB-2

Python code to convert a colored Image to Gray-Scale Image.

PROGRAM:

```
import numpy as np
import cv2
# Load images
img_color = cv2.imread('./image/image.jpg', cv2.IMREAD_COLOR)
img_gray = cv2.imread('./image/image.jpg', cv2.IMREAD_GRAYSCALE)
# Check if the images are loaded successfully
if img_color is None:
    print("Error: Could not load color image './image/image.jpg'")
if img_gray is None:
    print("Error: Could not load grayscale image './image/image.jpg'")
# Set desired width and height
desired_width = 300 # Example width
desired_height = 200 # Example height
# Resize images if they are loaded correctly
if img_color is not None:
    img_color= cv2.resize(img_color, (desired_width, desired_height))
if img_gray is not None:
    img_gray= cv2.resize(img_gray, (desired_width, desired_height))
# Display resized images
if img_color is not None:
    cv2.imshow('Color Image', img_color)
if img_gray is not None:
    cv2.imshow('Grayscale Image', img_gray)
# Wait for a key press and close windows
cv2.waitKey(0)
cv2.destroyAllWindows()
# Print shape and first channel of the resized color image
if img_color is not None:
    print("Color image shape:", img_color.shape)
    print("First channel (Blue) of color image:\n", img_color[:, :, 0])
```

OUTPUT



LAB-3
Python code to convert a circle into triangle and rectangle.

PROGRAM
```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation

# Function to generate circle points
def circle_points(radius, num_points=100):
    theta = np.linspace(0, 2*np.pi, num_points)
    x = radius * np.cos(theta)
    y = radius * np.sin(theta)
    return x, y

# Function to generate triangle points
def triangle_points(base, height, num_points=100):
    x = np.zeros(num_points)
    y = np.zeros(num_points)
    third = num_points // 3
    x[:third] = np.linspace(-base/2, base/2, third)
    y[:third] = -height / 2
    x[third:2*third] = np.linspace(base/2, 0, third)
    y[third:2*third] = np.linspace(-height/2, height, third)
    x[2*third:] = np.linspace(0, -base/2, num_points - 2*third)
    y[2*third:] = np.linspace(height, -height/2, num_points - 2*third)
    return x, y

# Function to generate rectangle points
def rectangle_points(width, height, num_points=100):
    num_points_side = num_points // 4
    x = np.concatenate([
        np.linspace(-width/2, width/2, num_points_side),
        np.full(num_points_side, width/2),
        np.linspace(width/2, -width/2, num_points_side),
        np.full(num_points_side, -width/2)
    ])
    y = np.concatenate([
        np.full(num_points_side, -height/2),
        np.linspace(-height/2, height/2, num_points_side),
        np.full(num_points_side, height/2),
        np.linspace(height/2, -height/2, num_points_side)
    ])
    return x, y

# Interpolation function
def interpolate_points(points1, points2, t):
    x1, y1 = points1
    x2, y2 = points2
    x = (1 - t) * x1 + t * x2
    y = (1 - t) * y1 + t * y2
    return x, y

# Initialize figure and axis
fig, ax = plt.subplots()
ax.set_xlim(-2, 2)
ax.set_ylim(-2, 2)
ax.set_aspect('equal')

# Generate shape points
circle = circle_points(1, num_points=100)
triangle = triangle_points(2, 2, num_points=100)
rectangle = rectangle_points(2, 1, num_points=100)

# Initialize plot
line, = ax.plot([], [], 'b-')

# Update function for animation
def update(frame):
    if frame < 100:
        t = frame / 100.0
        x, y = interpolate_points(circle, triangle, t)
    elif frame < 200:
        t = (frame - 100) / 100.0
        x, y = interpolate_points(triangle, rectangle, t)
    else:
        x, y = rectangle
    line.set_data(x, y)
    return line,

# Create animation
ani = animation.FuncAnimation(fig, update, frames=np.arange(0, 300), blit=True,
interval=50)
plt.show()
```

OUTPUT

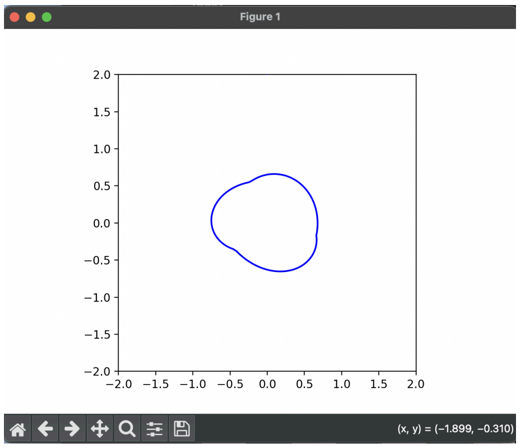


Figure 1
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0 −1.5 −1.0 −0.5 0.0 0.5 1.0 1.5 2.0
(x, y) = (−1.899, −0.310)

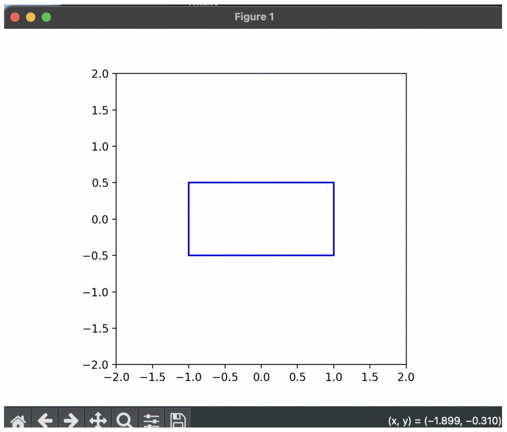


Figure 1
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0 −1.5 −1.0 −0.5 0.0 0.5 1.0 1.5 2.0
(x, y) = (−1.899, −0.310)

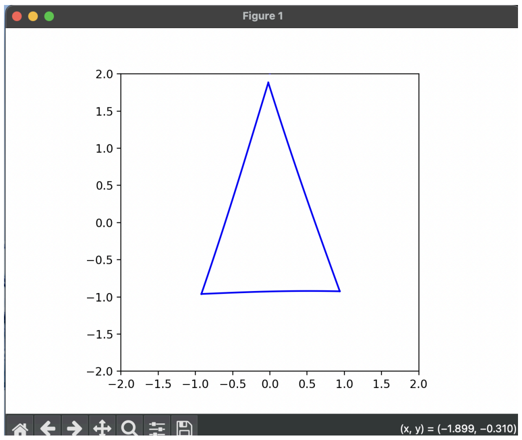


Figure 1
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0 −1.5 −1.0 −0.5 0.0 0.5 1.0 1.5 2.0
(x, y) = (−1.899, −0.310)

LAB:4

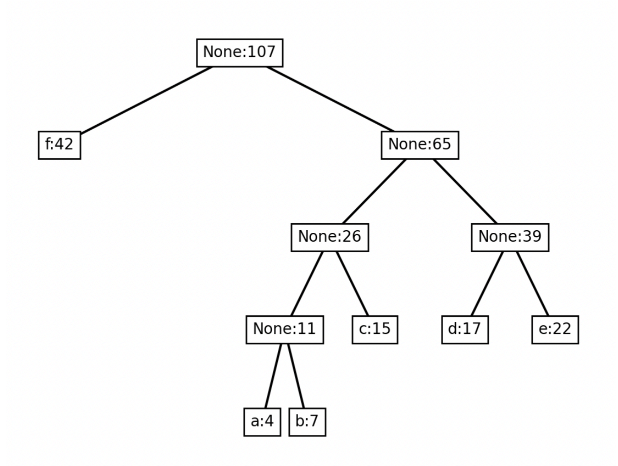
Program to simulate Huffman Code through Python Code.

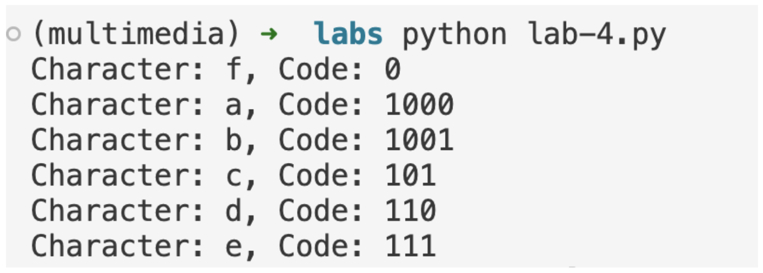
PROGRAM:

```
import heapq
import matplotlib.pyplot as plt
class Node:
    def __init__(self, symbol=None, frequency=None):
        self.symbol = symbol
        self.frequency = frequency
        self.left = None
        self.right = None
    def __lt__(self, other):
        return self.frequency < other.frequency
def build_huffman_tree(chars, freq):
    priority_queue = [Node(char, f) for char, f in zip(chars, freq)]
    heapq.heapify(priority_queue)
    while len(priority_queue) > 1:
        left_child = heapq.heappop(priority_queue)
        right_child = heapq.heappop(priority_queue)
        merged_node = Node(frequency=left_child.frequency +
right_child.frequency)
        merged_node.left = left_child
        merged_node.right = right_child
        heapq.heappush(priority_queue, merged_node)
    return priority_queue[0]
def generate_huffman_codes(node, code="", huffman_codes={}):
    if node is not None:
        if node.symbol is not None:
            huffman_codes[node.symbol] = code
        generate_huffman_codes(node.left, code + "0", huffman_codes)
        generate_huffman_codes(node.right, code + "1", huffman_codes)
    return huffman_codes
def visualize_huffman_tree(node, ax=None, x=0, y=0, dx=1, dy=1):
    if node is not None:
        if ax is None:
            fig, ax = plt.subplots()
        ax.text(x, y, f"{node.symbol}:{node.frequency}", ha="center",
va="center",
                bbox=dict(facecolor='white', edgecolor='black'))
        if node.left is not None:
            ax.plot([x, x-dx], [y, y-dy], 'k-')
            visualize_huffman_tree(node.left, ax, x-dx, y-dy, dx/2, dy)
        if node.right is not None:
            ax.plot([x, x+dx], [y, y-dy], 'k-')
            visualize_huffman_tree(node.right, ax, x+dx, y-dy, dx/2, dy)
chars = ['a', 'b', 'c', 'd', 'e', 'f']
freq = [4, 7, 15, 17, 22, 42]
root = build_huffman_tree(chars, freq)
huffman_codes = generate_huffman_codes(root)
```

```
for char, code in huffman_codes.items():
    print(f"Character: {char}, Code: {code}")
fig, ax = plt.subplots()
visualize_huffman_tree(root, ax)
ax.axis('off')
plt.show()
```

OUTPUT:



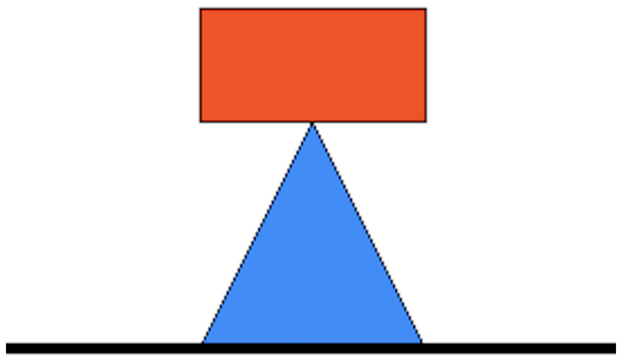


```
○ (multimedia) ➜  labs python lab-4.py
Character: f, Code: 0
Character: a, Code: 1000
Character: b, Code: 1001
Character: c, Code: 101
Character: d, Code: 110
Character: e, Code: 111
```

LAB:5
Python Program to draw logo.
PROGRAM:

```
from PIL import Image,ImageDraw,ImageFont
width,height=600,400
image=Image.new("RGB",(width,height),"white")
draw=ImageDraw.Draw(image)
try:
    font=ImageFont.truetype("arial.ttf",40)
except IOError:
    font=ImageFont.load_default()
black="black"
blue=(30,144,255)
red=(255,69,0)
triangle=[(width//2,height//4),(width//2-50,height//4+100),(widt
h//2+50,height//4+100)]
draw.polygon(triangle,outline=black,fill=blue)
rectangle_top_left=(width//2-50,height//4-50)
rectangle_bottom_right=(width//2+50,height//4)
draw.rectangle([rectangle_top_left,rectangle_bottom_right],outli
ne=black,fill=red)
line_start=(0,height//2)
line_end=(width,height//2)
draw.line([line_start,line_end],fill=black,width=5)
text="New Logo"
left, top, right, bottom = draw.textbbox((0, 0), text,
font=font)
text_width = right - left
text_height = bottom - top
text_x=(width-text_width)//2
text_y=height-text_height-50
draw.text((text_x,text_y),text,fill=black,font=font)
image.save("new_logo.png")
image.show()
```

OUTPUT

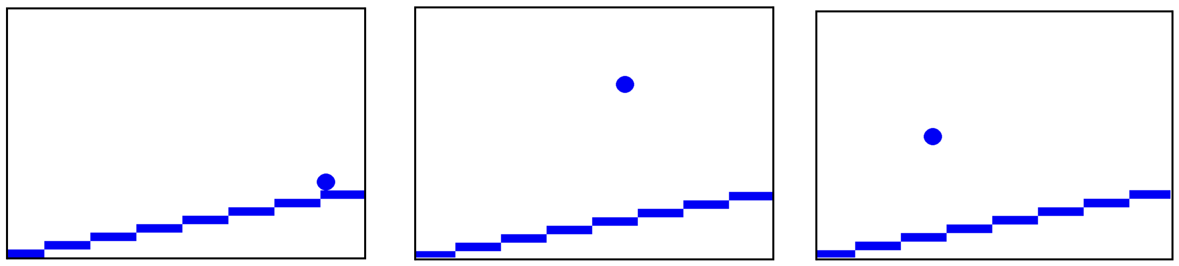


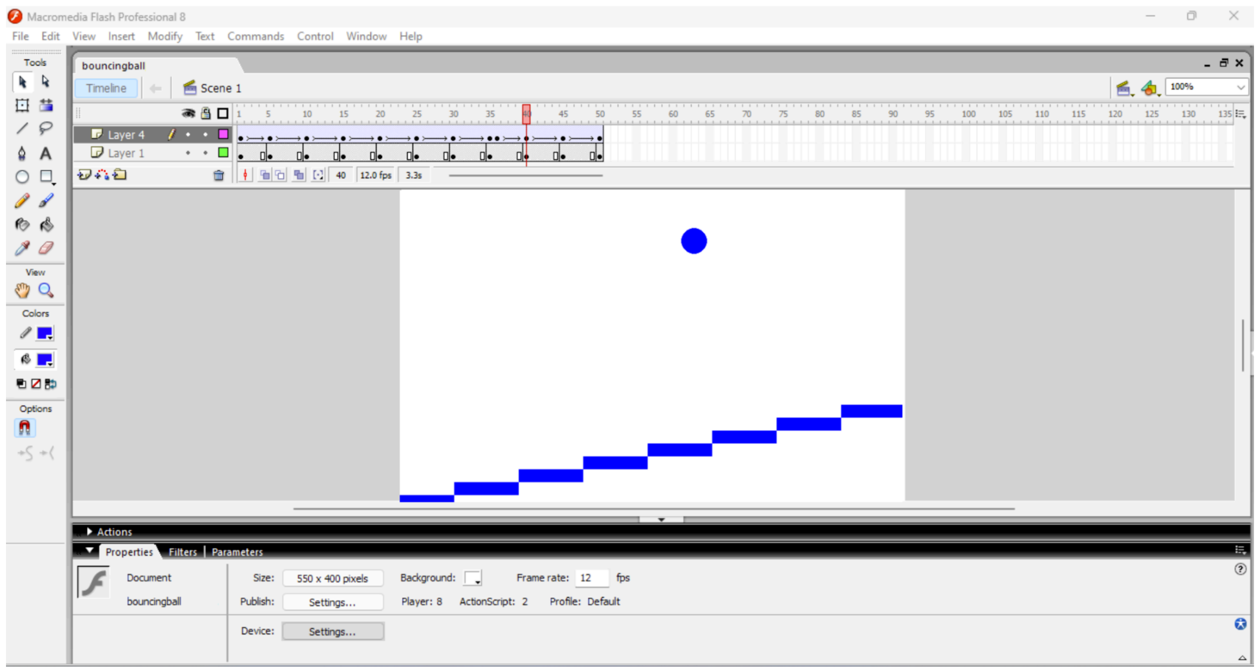
New Logo

LAB:6
Python Program to draw logo.
PROGRAM:

Create an animation to indicate a ball bouncing on steps.
STEPS:
Step 1: Go to start –macromedia- click on flash document
Step 2: Select the line tool and draw the steps, color it using the paint bucket tool Step 3: Select the circle from the tool bar and create a circle on the work area
Step 4: Now fill the color to the circle using the paint bucket tool from the tool bar Step 5: Go to frames right click on the first frame and choose insert key frame.
Slightly move the ball. Repeat the same procedure by adding new key frame to show the ball change the shape of the ball slightly when it touches the surface.
Step 6: In order to change the shape use the free transform tool
Step 7: Go to control and click on test movies, you will observe the ball bouncing on steps

Output:



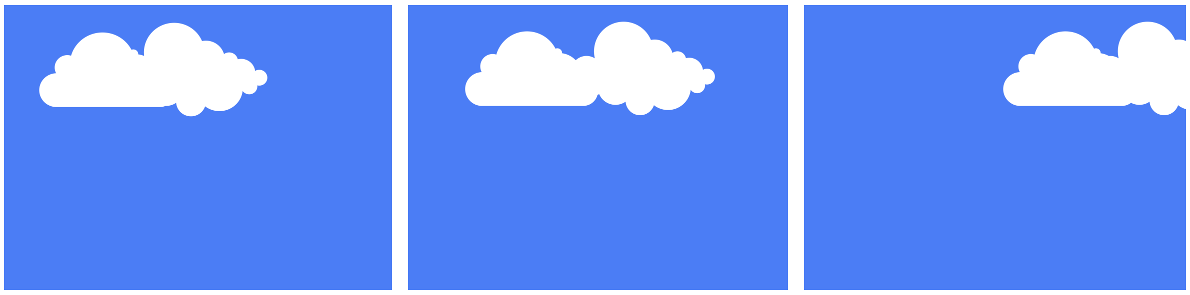

Macromedia Flash Professional 8
File Edit View Insert Modify Text Commands Control Window Help
Tools
View
Colors
Options
bouncingball
Timeline
Scene 1
Layer 4
Layer 1
Actions
Properties Filters Parameters

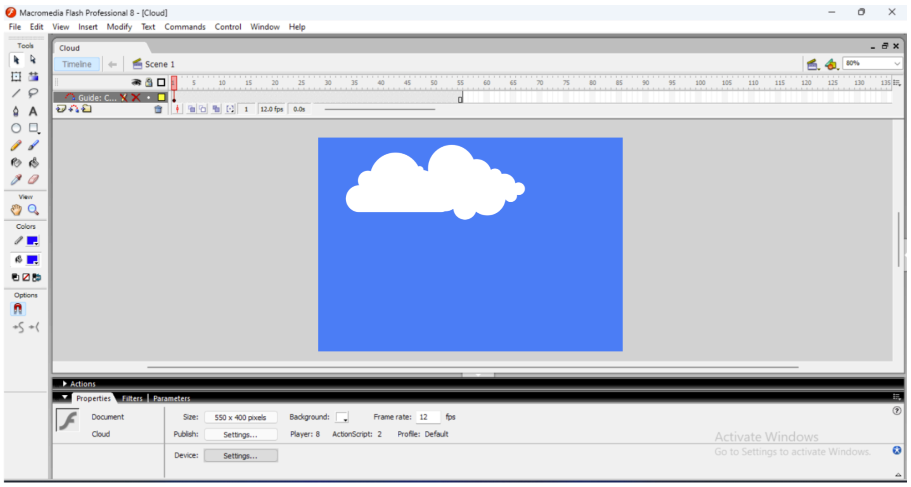
Document
bouncingball
Size: 550 x 400 pixels
Background:
Frame rate: 12 fps
Publish: Settings...
Player: 8 ActionScript: 2 Profile: Default
Device: Settings...

LAB:7
Create an animation to simulate the movement of a cloud.

STEPS:
Step 1: Go to start –macromedia- click on flash document
Step 2: Create a blue background in layer 1
Step 3: Now insert a layer 2 and draw the clouds in this layer
Step 4: In order to create the clouds, go to tool bar and select brush option, draw the cloud in layer 2
Step 5: Fill the color to the cloud, right click on it- choose convert to symbol option- give the name as cloud
Step 6: Select the movie clip option and click ok
Step 7: Go to filter color to white
click on the + symbol select glow to apply glowing effect select the
Step 8: Under glow and adjust the blur x / blur y values. [x=10, y=10] Step 9: Give the appropriate blur effect to the cloud
Step 10: Go to frames, insert keyframe on both the layer, create the motion tween on 2nd layer and move the clouds
Step 11: Finally go to control click on test movies

Output:




Macromedia Flash Professional 8 - [Cloud]
File Edit View Insert Modify Text Commands Control Window Help
Tools
View
Colors
Options
Cloud
Timeline
Scene 1
Guide: C...
Actions
Properties Filters Parameters

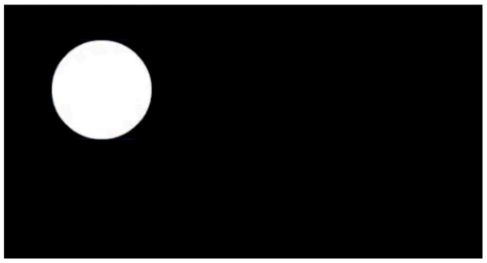
Document
Cloud

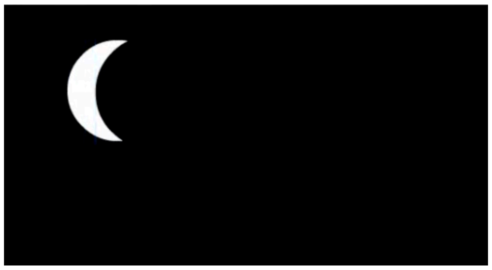
Size: 550 x 400 pixels Background: Frame rate: 12 fps
Publish: Settings... Player: 8 ActionScript: 2 Profile: Default
Device: Settings...

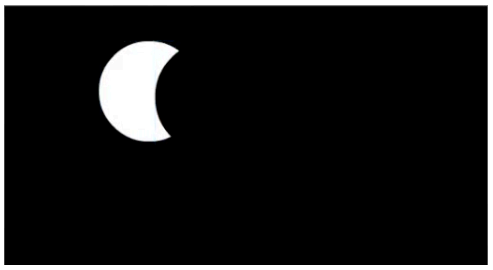
Activate Windows
Go to Settings to activate Windows.

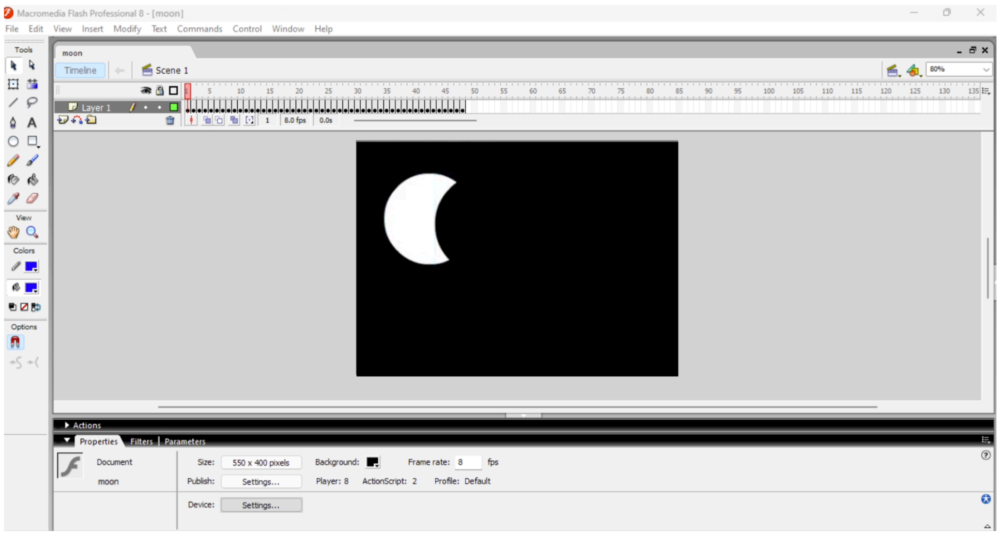
LAB:8
Create an animation to represent the growing moon

STEPS:
Step 1: Open flash 8 software select the properties tool click on flash document choose the background to black
go to windows properties
Step 2: Go to the fill color under the toolbar select the white color
Step 3: Select the oval tool in order to draw the moon, you will get a white circle
Step 4: Select the white circle on the worksheet using the selection tool to symbol select movie clip give suitable name eg: moon click ok
right click convert
Step 5: Go to filter click on the + symbol select glow to apply glowing effect color to white under the glow and adjust the blur x / blur y values. [x=12, y=12]
select the
Step 6: Click on the + symbol again and chose blur again adjust the blur x / blur y values
Step 7: Place the moon wherever you want on the work area, double click on layer 1 and rename as MOON
Step 8: Insert another layer rename it as Animation
Step 9: Select the fill color to black select oval tool and draw a circle on the moon to cover the moon select the newly added circle right click convert to symbol movie clip
name it as Animation
Step 10: Go to filter select + symbol give the glow and blur effect as did for moon
Step 11: Select the 150th frame in moon layer same for animation layer
right click insert keyframe. Repeat the
Step 12: Click on the 149th keyframe of animation layer right click select the animation movie clip and move slowly across the moon
press create motion
Step 13: Finally go to the control test movie and you will get a growing moon as the output.









Macromedia Flash Professional 8 - [moon]
File Edit View Insert Modify Text Commands Control Window Help
Tools
View
Colors
Options
moon
Timeline
Scene 1
80%
Layer 1
Actions
Properties
Filters
Parameters

Document moon

Size: 550 x 400 pixels Background: Frame rate: 8 fps

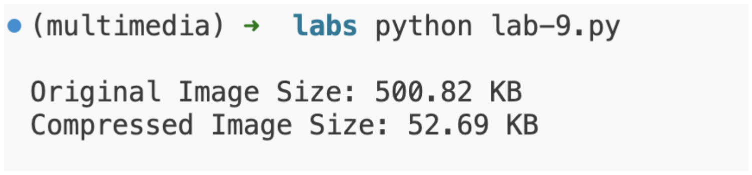
Publish: Settings... Player: 8 ActionScript: 2 Profile: Default

Device: Settings...

1 8.0 fps 0.0s

LAB:9
WAP in python to compress the image.

PROGRAM
```
from PIL import Image
import os
def compress_image(input_image_path, output_image_path, quality=85):
try:
img = Image.open(input_image_path)
img.save(output_image_path, "JPEG", quality=quality, optimize=True)
original_size = os.path.getsize(input_image_path)
compressed_size = os.path.getsize(output_image_path)
print(f"Original Image Size: {original_size / 1024:.2f} KB")
print(f"Compressed Image Size: {compressed_size / 1024:.2f} KB")
except Exception as e:
print(f"An error occurred: {e}")
def main():
input_image_path = "./image/logo.png"
output_image_path = "compressed_image.jpg"
compression_quality = 60
compress_image(input_image_path, output_image_path,
quality=compression_quality)
if __name__ == "__main__":
main()
```

OUTPUT

```
● (multimedia) ➜  labs python lab-9.py

  Original Image Size: 500.82 KB
  Compressed Image Size: 52.69 KB
```

Original **Compressed**