

```
from math import pi, cos, sin
import math
import turtle
import tkinter as tk
from tkinter import messagebox
import winsound

def start_create_window():
    root.configure(bg="lightblue")

    image = tk.PhotoImage(file="C:\\Users\\RUCHIKA\\Desktop\\pic.png")
    image = image.subsample(2,2)
    img_label = tk.Label(root, image=image, bg="lightblue")
    img_label.image = image
    img_label.pack(pady=30)

    start_button = tk.Button(root, text="Start", command=show_input_window,
                             bg="gold", font=("Arial", 14))
    start_button.pack()

def show_input_window():
    clear_widgets()
    bg_image = tk.PhotoImage(file="C:\\Users\\RUCHIKA\\Desktop\\14.png")
    bg_label = tk.Label(root, image=bg_image)
    bg_label.image = bg_image
    bg_label.place(x=0, y=0, relwidth=1, relheight=1)

    lbl = tk.Label(root, text="Enter the number of sides:", font=("Arial", 14))
    lbl.pack()

    entry = tk.Entry(root, width=6, font=("Arial", 14))
    entry.pack()

    next_button = tk.Button(root, text="Next", command=lambda: process_input
                             (entry), bg="gold", font=("Arial", 14))
    next_button.pack()

def process_input(entry):
    global no_sides
    sides = entry.get().strip()
    if sides:
        no_sides = int(sides)
        if no_sides >= 3:
            create_side_input_window()
        else:
            lbl2 = tk.Label(root, text="Enter a valid number!!!!", font=
```

```
        ("Arial", 14))
        lbl2.pack()
    else:
        lbl2 = tk.Label(root, text="Enter the number of sides!", font=("Arial", 14))
        lbl2.pack()

def create_side_input_window():
    clear_widgets()

    bg_image = tk.PhotoImage(file="C:\\Users\\RUCHIKA\\Desktop\\14.png")
    bg_label = tk.Label(root, image=bg_image)
    bg_label.image = bg_image
    bg_label.place(x=1, y=1, relwidth=1, relheight=1)

    lbl = tk.Label(root, text="Enter the individual side lengths of the polygon:", font=("Arial", 14))
    lbl.pack()

    global side_entries
    side_entries = []
    for i in range(no_sides):
        lbl_side = tk.Label(root, text=f"Side {i + 1}:", font=("Arial", 14))
        lbl_side.pack()
        side_entry = tk.Entry(root, width=6, font=("Arial", 14))
        side_entry.pack()
        side_entries.append(side_entry)

    submit_button = tk.Button(root, text="Submit", command=calculate_area, bg="gold", font=("Arial", 14))
    submit_button.pack()

    lbl.lift()
    for entry in side_entries:
        entry.lift()
    submit_button.lift()

def calculate_area():
    global sidelengths
    sidelengths = [float(entry.get()) for entry in side_entries]
    area = polygon_area(sidelengths)
    label = tk.Label(root, text=f"The Maximum area of the given polygon is {area:.2f}", font=("Arial", 14, "bold"))
    label.pack()
```

```
next_button = tk.Button(root, text="Next", command=draw_polygon, bg="gold",
font=("Arial", 14))
next_button.pack()

def polygon_area(side_lengths):
    if len(side_lengths) < 3:
        raise ValueError("A polygon must have at least 3 sides.")

    n = len(side_lengths)

    x_coords = [0]
    y_coords = [0]

    for i in range(n):
        x_coords.append(x_coords[-1] + side_lengths[i] * cos(2 * pi * i / n))
        y_coords.append(y_coords[-1] + side_lengths[i] * sin(2 * pi * i / n))

    area = 0
    for i in range(n):
        area += x_coords[i] * y_coords[i + 1] - x_coords[i + 1] * y_coords[i]

    return abs(area) / 2

def calculate_vertices(sides, side_lengths):
    vertices = []
    angle_increment = 360 / sides
    current_angle = 0

    for length in side_lengths:
        x = length * math.cos(math.radians(current_angle))
        y = length * math.sin(math.radians(current_angle))
        vertices.append((x, y))
        current_angle += angle_increment

    return vertices

def draw_polygon():
    clear_widgets()
    turtle.speed(1)
    turtle.bgcolor("pink")
    turtle.color("navy blue")
    turtle.title('Polygon')
    turtle.pen(fillcolor="red", pencolor="navy blue", pensize=6)

    vertices = calculate_vertices(no_sides, sidelengths)

    turtle.penup()
```

---

```
turtle.goto(vertices[0])
turtle.pendown()

for vertex in vertices:
    turtle.goto(vertex)

turtle.goto(vertices[0])

winsound.Beep(2500, 500)
turtle.hideturtle()

exit_button = tk.Button(root, text="Exit", command=exit_application,
    bg="red", font=("Arial", 14))
exit_button.pack()

def clear_widgets():
    for widget in root.winfo_children():
        widget.destroy()

def exit_application():
    if messagebox.askyesno("Exit", "Do you want to exit?"):
        root.destroy()

if __name__ == "__main__":
    root = tk.Tk()
    root.title("Polygon Generation")
    root.state("zoomed")
    start_create_window()
    root.mainloop()
```