PYTHON TKINTER ASSINMEN

1.Write a Python program that implements a seating reservation system for an auditorium, arranging seat buttons in a grid layout.

CODE-

```
import tkinter as tk

class AuditoriumReservationSystem:

    def __init__(self, parent):

        self.parent = parent

        self.parent.title("Auditorium Reservation")

        # Auditorium layout (number of rows and columns)

        self.rows = 5

        self.columns = 6

        # Create a Frame to hold seat buttons

        self.seating_frame = tk.Frame(parent)

        self.seating_frame.pack()

        # Create and display seat buttons in a grid

        self.create_seats()

    def create_seats(self):

        self.seat_buttons = []

        for row in range(self.rows):

            seat_row = []

            for col in range(self.columns):

                seat_button = tk.Button(self.seating_frame, text=f"Seat {row+1}-{col+1}", width=8,
height=2, command=lambda r=row, c=col: self.reserve_seat(r, c))

                seat_button.grid(row=row, column=col, padx=5, pady=5)

                seat_row.append(seat_button)

            self.seat_buttons.append(seat_row)


    def reserve_seat(self, row, col):

        seat_button = self.seat_buttons[row][col]
```

```
        if seat_button['state'] == 'normal':
            seat_button.configure(bg='lightgreen', state='disabled')
            print(f"Reserved Seat {row+1}-{col+1}")
        else:
            print(f"Seat {row+1}-{col+1} is already reserved.")
if __name__ == "__main__":
    parent = tk.Tk()
    app = AuditoriumReservationSystem(parent)
    parent.mainloop()
```

2. Write a Python program that implements a contact information form with labels and Entry widgets. Organize them within a Frame widget using the Place geometry manager.

CODE-

```
import tkinter as tk

class ContactForm:

    def __init__(self, parent):

        self.parent = parent

        self.parent.title("Contact Information Form")

        # Create a Frame to hold the form elements

        self.form_frame = tk.Frame(parent, padx=20, pady=20)

        self.form_frame.pack()



        # Create labels and Entry widgets for the contact form

        self.name_label = tk.Label(self.form_frame, text="Name:")
```

```
        self.name_label.grid(row=0, column=0, sticky="w")

        self.name_entry = tk.Entry(self.form_frame)

        self.name_entry.grid(row=0, column=1)

        self.email_label = tk.Label(self.form_frame, text="Email:")

        self.email_label.grid(row=1, column=0, sticky="w")

        self.email_entry = tk.Entry(self.form_frame)

        self.email_entry.grid(row=1, column=1)

        self.phone_label = tk.Label(self.form_frame, text="Phone:")

        self.phone_label.grid(row=2, column=0, sticky="w")

        self.phone_entry = tk.Entry(self.form_frame)

        self.phone_entry.grid(row=2, column=1

        # Create a Submit button

        self.submit_button = tk.Button(self.form_frame, text="Submit",
command=self.submit_form)

        self.submit_button.grid(row=3, columnspan=2, pady=10)

    def submit_form(self):

        # Retrieve and display the submitted data

        name = self.name_entry.get()

        email = self.email_entry.get()

        phone = self.phone_entry.get()

        # Display the submitted data in a message box

        message = f"Name: {name}\nEmail: {email}\nPhone: {phone}"
```

```
        tk.messagebox.showinfo("Submitted Information", message)


if __name__ == "__main__":
    parent = tk.Tk()
    app = ContactForm(parent)
    parent.mainloop()
```

3. Write a Python program that creates a basic calendar application with labels for days of the week and dates. Use the Grid geometry manager to arrange the labels.

CODE-

```
import tkinter as tk
import calendar
# Function to update the calendar
def update_calendar(year, month):
    cal_text.config(text=calendar.month(year, month))
# Create the main window
root = tk.Tk()
root.title("Calendar App.")
# Create labels for dates using the Grid geometry manager
current_year = 2000
current_month = 2
cal_text = tk.Label(root, text="")
cal_text.grid(row=0, column=1, columnspan=7, padx=11, pady=11)
update_calendar(current_year, current_month)
# Create "Previous" and "Next" buttons to navigate the calendar
def prev_month():
    global current_year, current_month
    current_month -= 1
```

```
        if current_month < 1:
            current_year -= 1
            current_month = 12
        update_calendar(current_year, current_month)

    def next_month():
        global current_year, current_month
        current_month += 1
        if current_month > 12:
            current_year += 1
            current_month = 1
        update_calendar(current_year, current_month)

    prev_button = tk.Button(root, text="Previous", command=prev_month)
    prev_button.grid(row=1, column=0, padx=10, pady=10)

    next_button = tk.Button(root, text="Next", command=next_month)
    next_button.grid(row=1, column=9, padx=10, pady=10)

    # Start the Tkinter event loop
    root.mainloop()
```

4. Write a Python GUI program to create three radio buttons widgets using tkinter module.

CODE-

```
import tkinter as tk
parent = tk.Tk()
parent.title("Radiobutton ")
parent.geometry('350x200')
radio1 = tk.Radiobutton(parent, text='First', value=1)
radio2 = tk.Radiobutton(parent, text='Second', value=2)
radio3 = tk.Radiobutton(parent, text='Third', value=3)
```

```
radio1.grid(column=0, row=0)
radio2.grid(column=1, row=0)
```

5. Write a Python GUI program to create a Menu bar with File, Edit, and Help menus, each containing submenu items using tkinter module.

CODE-

```
import tkinter as tk
from tkinter import Menu


def new_file():
    print("New File")


def open_file():
    print("Open File")


def save_file():
    print("Save File")


def cut_text():
    print("Cut Text")


def copy_text():
    print("Copy Text")
```

```
def paste_text():
    print("Paste Text")

def about():
    print("About this application")

parent = tk.Tk()
parent.title("Tkinter Application")

# Create a menu bar
menu_bar = Menu(parent)
parent.config(menu=menu_bar)

# Create File menu and submenu items
file_menu = Menu(menu_bar, tearoff=0)
menu_bar.add_cascade(label="File", menu=file_menu)
file_menu.add_command(label="New", command=new_file)
file_menu.add_command(label="Open", command=open_file)
file_menu.add_command(label="Save", command=save_file)
file_menu.add_separator()
file_menu.add_command(label="Exit", command=parent.quit)
```

```
# Create Edit menu and submenu items
edit_menu = Menu(menu_bar, tearoff=0)
menu_bar.add_cascade(label="Edit", menu=edit_menu)
edit_menu.add_command(label="Cut", command=cut_text)
edit_menu.add_command(label="Copy", command=copy_text)
edit_menu.add_command(label="Paste", command=paste_text)
# Create Help menu and submenu items
help_menu = Menu(menu_bar, tearoff=0)
menu_bar.add_cascade(label="Help", menu=help_menu)
help_menu.add_command(label="About", command=about)
parent.mainloop()
```

6. Write a Python GUI program to create a Treeview widget with columns and sorting functionality using tkinter module.

CODE-

```
import tkinter as tk
from tkinter import ttk


# Function to sort the Treeview by column
def sort_treeview(tree, col, descending):
    data = [(tree.set(item, col), item) for item in tree.get_children('')]
    data.sort(reverse=descending)
    for index, (val, item) in enumerate(data):
        tree.move(item, '', index)
```

```
    tree.heading(col, command=lambda: sort_treeview(tree, col, not descending))


parent = tk.Tk()
parent.title("Sortable Treeview Example")


# Create a Treeview widget with columns
columns = ("Name", "Age", "Country")
tree = ttk.Treeview(parent, columns=columns, show="headings")


# Configure column headings and sorting functionality
for col in columns:
    tree.heading(col, text=col, command=lambda c=col: sort_treeview(tree, c, False))
    tree.column(col, width=100)


# Insert data into the Treeview
data = [("Taliesin Megaira", 35, "USA"),
        ("Fionn Teofilo", 28, "UK"),
        ("Ata Caishen", 45, "France"),
        ("Marina Evie", 38, "Spain"),
        ("Xanthe Eligio", 32, "France"),
        ("Fausta Tone", 30, "Australia"),
        ("Yawan Idane", 42, "Mongolia"),
        ("Rolph Noah", 30, "South Africa")]


for item in data:
    tree.insert('', 'end', values=item)
tree.pack()
parent.mainloop()
```