

EXPERIMENT-1

Aim - Implement all major functions of string.h in a single C program using switch case to select specific functions from the user

- Strlen - Find length of string
- Strcat - Concatenate 2 strings
- Strcpy - Copy string1 in string2
- Strcmp - Compare 2 strings
- Strrev - Reverse a string

Theory - There are various standard library functions and a macro defined under string.h to manipulate and perform operations on strings and array of characters in C programming

The functions are explained above.

Code

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
```

```

int main() {
    int q;
    char a[100], b[100];
    cout << "Enter l1: Length of the string l2: compare
           strings l3: concatenate two strings l4:
           Reverse a string l5: copy a string l6 ";
    cin >> q;
    switch(q) {

```

Case 1:

```

        cout << "Input a String : ";
        cin >> a;
        cout << "String Length is " << strlen(a);
        break;
    }

```

Case 2:

```

    cout << "Input 2 String : ";
    cin >> a >> b;
    if(strcmp(a, b) == 0)
        cout << "Strings Equal";
    else
        cout << "Strings unequal";
    break;
}

```

Case 3:

```

    cout << "Input 2 String : ";
}

```

OUTPUT

- 1: Length of the String
- 2: Compare Strings
- 3: Concatenate two strings
- 4: Reverse a string
- 5: Copy a string

Input two string : good
morning

good morning

Scarf (" / s / - s ", a, b);
strcat (a, b);
hunry (" / s ", a);
break;

Case 4:

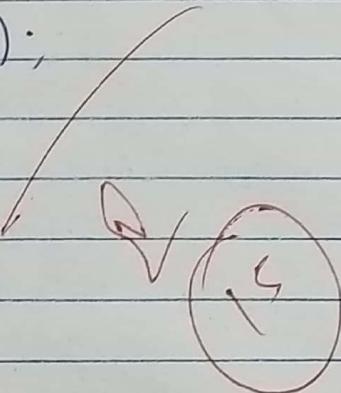
hunry (" Input a string ");
Scarf (" / s ", a);
hunry (strrev(a));
break;

Case 5:

hunry (" Input2 string ");
Scarf (" / . s / - s , a, b");
strcat (a, b);
hunry (" / s ", b);
}

return 0;

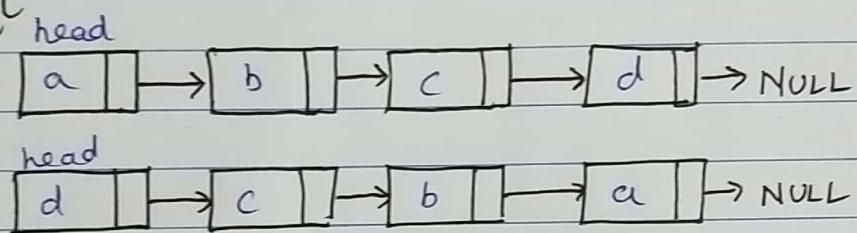
}



EXPERIMENT-2

Aim - To reverse a linked list iteratively as well as recursively.

THEORY:



ITERATIVE METHOD -

- Initialize three pointers `prev` as null, `curr` as `head` and `next` as null.
- Iterate through the linked list. In loop, do the following
 - // Before changing next of current
 - // Store next of mode
 - $\text{next} = \text{curr} \rightarrow \text{next}$
 - // change next of current
 - $\text{curr} \rightarrow \text{next} = \text{prev}$
 - // Move prev & curr one step forward
 - $\text{prev} = \text{curr}$
 - $\text{curr} = \text{next}$

RECURSIVE METHOD :-

- Create a function reverse
- In function do the following
 - if ($h \rightarrow \text{next} = \text{NULL}$)
 $\text{head} = h;$
 - Reverse ($h \rightarrow \text{next}$)

PROGRAM

```
# include <stdio.h>
# include <stdlib.h>
```

```
Struct Node {
    int data;
    Struct Node* next;
};
```

```
Static void reverse (Struct Node** head_ref)
```

```
{
```

```
Struct Node* prev = NULL;
```

```
Struct Node* current = *head_ref;
```

```
Struct Node* next = NULL;
```

```
while (current != NULL) {
```

```
    next = current -> next;
```

```
    next -> current -> next = prev;
```

```
hrev = current;  
current = next;  
* head->ref = hrev;
```

RECURSIVE Approach:-

```
Struct LinkedList {
```

```
    Node* head;
```

```
    LinkedList();
```

```
    }
```

```
    head = NULL;
```

```
}
```

```
    Node* reverse (Node* head)
```

```
{
```

```
    if (head == NULL || head->next == NULL)
```

```
        return head;
```

```
    Node* rest = reverse (head->next);
```

```
    head->next->next = head;
```

```
    head->next = NULL;
```

```
    return rest;
```

```
}
```

RESULT - Linklist reversal has done successfully

PRACTICAL - 3

Aim - Write a program in C to implement iterative tower of Hanoi.

Theory - Tower of Hanoi is a mathematical puzzle. It consists of 3 holes and a number of disks of different sizes which can slide onto any holes.

The objective of the puzzle is to move all disks from one hole to another hole with the help of a third hole called the Auxiliary hole.

The puzzle has 2 rules :-

- You can't place a larger disk on a smaller disk
- only one disk can be moved at a time.

Code -

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <limits.h>
```

Struct Stack {

 unsigned capacity;

 int top;

 int * array;

}

Struct Stack* createStack (unsigned capacity)

{

 Struct Stack* stack = (struct Stack*) malloc(sizeof(Struct
 Stack));

 stack → capacity = capacity;

 stack → Top = 1;

 stack → array = (int*) malloc(stack → capacity * sizeof(int));
 return stack;

}

int IsFull (Struct Stack* stack)

{

 return (stack → top == stack → capacity - 1);

y

int IsEmpty (Struct Stack* stack)

{

 if (IsFull (stack))

 return (stack → top == -1);

y

```
void push (Struct Stack *Stack, int item)
{
    if (isFull (Stack))
        return;
    Stack -> array [Stack -> top] = item;
}
```

```
int hoh (Struct Stack *Stack)
```

```
{
    if (isEmpty (Stack))
        return INT-MIN;
    return Stack -> array [Stack -> top -];
}
```

```
void moveDisk (char fromPeg, char toPeg, int disk)
```

```
{
    cout << "Move disk " << disk << " from " << fromPeg << " to " << toPeg << endl;
```

```
void moveDisks Between2Poles (Struct Stack *src, Struct Stack *dest, char s, char d)
```

```
{
    int hole1TopDisk = hoh (src);
    int hole2TopDisk = hoh (dest);

    if (hole1TopDisk == INT-MIN)
        push (src, hole2TopDisk);
    moveDisk (s, d, hole2TopDisk);
}
```

else if ($\text{hole}_2 \text{TohDisk} = \text{INT-MIN}$)

$\text{push}(\text{dest}, \text{hole}_1 \text{TohDisk});$

$\text{moveDisk}(\text{s}, \text{d}, \text{hole}_1 \text{TohDisk});$

else if ($\text{hole}_1 \text{TohDisk} > \text{hole}_2 \text{TohDisk}$)

$\text{push}(\text{src}, \text{hole}_1 \text{TohDisk});$

$\text{push}(\text{src}, \text{hole}_2 \text{TohDisk});$

$\text{moveDisk}(\text{d}, \text{s}, \text{hole}_2 \text{TohDisk});$

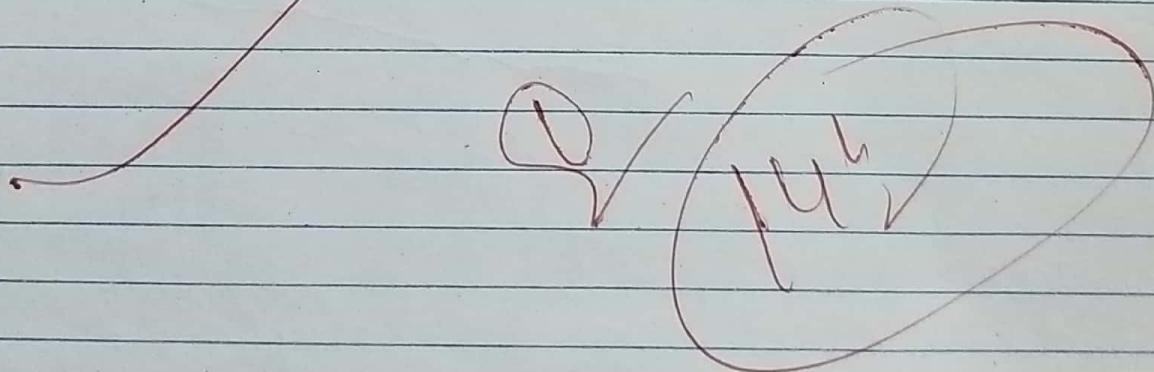
else

$\text{push}(\text{dest}, \text{hole}_2 \text{TohDisk});$

$\text{push}(\text{dest}, \text{hole}_1 \text{TohDisk});$

$\text{moveDisk}(\text{s}, \text{d}, \text{hole}_1 \text{TohDisk});$

3



PRACTICAL - 4

Aim - write a program in C++ to count the number of object of a class with help of static data member, function and constructor.

Theory - Count number of objects are created from a specific class using some static member functions. The static members are class properties, not the object parameters. For a single class there will only be one instance for static members.

When a new object is created, so the constructor will be called. Inside the constructor, the count value is increased. Thus we can get the output.

Code -

```
#include <iostream>
using namespace std;
class Test {
    int code;
    static int count;
public:
```

```
void setcode ( void )
{
    code = ++ count;
}

void showcode ( void )
{
    cout << "object number: " << code;
}

static void Showcount ( void )
{
    cout << "Count" << count;
}
```

```
int test :: count;
int main ()
{
    test t1, t2;
    t1.setcode();
    t2.setcode();
    test :: Showcount();
    test t3;
    t3.setcode();

    'test :: Showcount();
    t1.showcode();
    t2.showcode();
    return 3
}
```

CONCLUSION: The code was implemented successfully

PRACTICAL – 5

AIM – WAP in C++ or Java to declare a class Time with data members mm for minutes, ss for seconds and hh for hours. Define a parameterized constructor to assign time to its objects. Add two time objects using member function and assign to third objects. Implement all possible cases of time.

SOFTWARE USED – Code::Blocks

SOURCE CODE –

```
#include <iostream>
using namespace std;

class Time
{
private:
    int hours;
    int minutes;
    int seconds;

public:
    void getTime(void);
    void displayTime(void);
    void convertIntoSeconds(void);
    void addTime(Time T1,Time T2);

};

void Time::getTime(void)
{
    cout << "Please Enter the Time:" << endl;
    cout << "Enter the Hours: ";      cin>>hours;
    cout << "Enter the Minutes: ";   cin>>minutes;
    cout << "Enter the Seconds: ";   cin>>seconds;
}

void Time::displayTime(void)
{
    cout << endl;
    cout << "The Time is : ";
    cout << hours << ":" << minutes << ":" << seconds << endl;
}

void Time::addTime(Time T1,Time T2)
{

    this->seconds=T1.seconds+T2.seconds;
    this->minutes=T1.minutes+T2.minutes + this->seconds/60;;
    this->hours= T1.hours+T2.hours + (this->minutes/60);
    this->minutes %=60;
```

```
    this->seconds %=60;
}

void Time::convertIntoSeconds(void)
{
    seconds = hours*3600 + minutes*60 + seconds;
    cout<<"The Seconds in total are : "<<seconds;
}

int main()
{
    Time T1,T2,T3;
    T1.getTime();
    T2.getTime();
    //add two times
    T3.addTime(T1,T2);
    T3.displayTime();
    T3.convertIntoSeconds();
    return 0;
}
```

OUTPUT –

```
Please Enter the Time:
Enter the Hours: 5
Enter the Minutes: 30
Enter the Seconds: 20
Please Enter the Time:
Enter the Hours: 3
Enter the Minutes: 25
Enter the Seconds: 30

The Time is : 8:55:50
The Seconds in total are : 32150

...Program finished with exit code 0
Press ENTER to exit console.[]
```

PRACTICAL – 6

AIM – WAP in C++ to define a class Complex to represent a set of all complex numbers. Overload “+” operator to add two complex numbers using member function of the class and overload “*” operator to multiply two complex numbers using the friend function of the class complex.

SOFTWARE USED – Code::Blocks

SOURCE CODE –

```
#include<iostream>
using namespace std;

class complex
{
int real,imag;

public:
void get()
{
cout<<"Enter real and imaginary part respectively ";
cin>>real>>imag;
}

complex operator+(complex t1)
{
complex temp;
temp.real = t1.real+ this->real;
temp.imag = t1.imag + this->imag;
return(temp);
}

friend complex operator*(complex,complex);

void display()
{
cout<<"The complex number is "<<real<<"+"<<imag<<"\n";
}

complex operator * (complex t1,complex t2)
{
complex temp;
temp.real = t1.real * t2.real;
temp.imag = t1.imag* t2.imag;
return(temp);
}

int main()
{
```

```
complex t1,t2,t3,t4;
t1.get();
t2.get();
t3=t1+t2;
t4=t1*t2;
t3.display();
t4.display();
return(0);
}
```

OUTPUT -

```
Enter real and imaginary part respectively 4 7
Enter real and imaginary part respectively 3 2
The complex number is7+i9
The complex number is12+i14

...Program finished with exit code 0
Press ENTER to exit console.[]
```

PRACTICAL – 7A

AIM – Write a program in to prepare a list of 10 questions and their answers.

SOFTWARE USED – VS Code, Chrome

SOURCE CODE – For HTML, CSS and JavaScript Files

index.html

```
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>Quiz</title>
    <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body style="background: pink;"></body>
    <div class="container">
        <div id="quiz">

            <div class="title">
                <p>Let's Take a Quiz</p>
            </div>
            <hr style="margin-bottom: 20px">

            <p id="question"></p>

            <div class="buttons">
                <button id="btn0"><span
id="choice0"></span></button>
                <button id="btn1"><span
id="choice1"></span></button>
                <button id="btn2"><span
id="choice2"></span></button>
                <button id="btn3"><span
id="choice3"></span></button>
            </div>

            <hr style="margin-top: 50px">
            <footer>
                <p id="progress">Question x of y</p>
                <p>&copy; 2020 Copyright, Quiz</p>
                <p>Made with &hearts; by Bipin Lala</p>
            </footer>
        </div>
    </div>
</div>

<script src="question.js"></script>

</body>
</html>
```

Questions.js

```
function Quiz(questions) {
    this.score = 0;
    this.questions = questions;
    this.questionIndex = 0;
}

Quiz.prototype.getQuestionIndex = function() {
    return this.questions[this.questionIndex];
}

Quiz.prototype.guess = function(answer) {
    if(this.getQuestionIndex().isCorrectAnswer(answer)) {
        this.score++;
    }

    this.questionIndex++;
}

Quiz.prototype.isEnded = function() {
    return this.questionIndex === this.questions.length;
}

function Question(text, choices, answer) {
    this.text = text;
    this.choices = choices;
    this.answer = answer;
}

Question.prototype.isCorrectAnswer = function(choice) {
    return this.answer === choice;
}

function populate() {
    if(quiz.isEnded()) {
        showScores();
    }
    else {
        // show question
        var element = document.getElementById("question");
        element.innerHTML = quiz.getQuestionIndex().text;

        // show options
        var choices = quiz.getQuestionIndex().choices;
        for(var i = 0; i < choices.length; i++) {
            var element = document.getElementById("choice" + i);
            element.innerHTML = choices[i];
            guess("btn" + i, choices[i]);
        }
    }
}
```

```

        showProgress();
    }
};

function guess(id, guess) {
    var button = document.getElementById(id);
    button.onclick = function() {
        quiz.guess(guess);
        populate();
    }
};

function showProgress() {
    var currentQuestionNumber = quiz.questionIndex + 1;
    var element = document.getElementById("progress");
    element.innerHTML = "Question " + currentQuestionNumber +
" of " + quiz.questions.length;
};

function showScores() {
    var gameOverHTML = "<h1>Result</h1>";
    gameOverHTML += "<h2 id='score'> Your scores: " +
quiz.score + "</h2>";
    var element = document.getElementById("quiz");
    element.innerHTML = gameOverHTML;
};

// create questions here
var questions = [
    new Question("Who among the following proposed Helio-
centric model of solar system?", ["Copernicus",
"Ptolemy", "Aristotle", "Strabo"], "Copernicus"),
    new Question("Who had propounded the planetary laws?",
["Newton", "Kepler", "Galileo", "Copernicus"], "Kepler"),
    new Question("Which one of the following elements occurs
most abundantly in our universe?", ["Hydrogen",
"Nitrogen", "Helium", "Oxygen"], "Hydrogen"),
    new Question("The stellar and solar source of energy is:",
["Nuclear Fusion", "Nuclear Fission", "Electromagnetic
Induction", "Electromotive Force"], "Nuclear Fusion"),
    new Question(" The device employed to measure the
diameters of stars and our galaxy (Milky Way) is called:",
["Photometer", "Barometer", "Viscometer", "Interferometer"],
"Interferometer"),
    new Question(" A dog sent in first flight to space was
named", ["Laika", "Jupiter", "Lucky", "Luna"], "Laika"),
];

```

```

        new Question(" The first artificial satellite sent in
space was", ["Sputnik 1", "Sputnik 2", "Explorer 1", "Explorer
2"], "Sputnik 1"),

        new Question(" Who among the following proposed steady
state theory with respect to universe?", ["Einstein",
"Laplace", "Newton", "Hoyle"], "Hoyle"),

        new Question(" Which of the following planet is an example
of Terrestrial planet?", ["Earth", "Jupiter", "Saturn",
"Neptune"], "Earth"),

        new Question(" Which of the following is the least dense
planet among all the planets?", ["Earth", "Uranus", "Jupiter",
"Saturn"], "Saturn")
]

// create quiz

var quiz = new Quiz(questions);

// display quiz
populate();

```

style.css

```

@font-face {
    font-family: "Product Sans";
    font-style: normal;
    font-weight: 400;
    src: local("Open Sans"), local("OpenSans"),
url(https://fonts.gstatic.com/s/productsans/v5/HYvgU2fE2nRJvZ5
JFAumwegdm0LZdjqr5-oayXSOefg.woff2)
        format("woff2");
}

body {
    font-family: "Product Sans";
    text-align: center;
    /* align-items: center; */
    margin: 0;
    padding: 0;
    /* background: lightcoral; */

}

.container {
    padding: 0;
    margin: 0;
    background:white;
    width: 90vw;

```

```
display: inline-block;
max-height: 90vh;
overflow: scroll;
border-radius: 2px;
margin: 1rem;
}

::-webkit-scrollbar {
display: none;
}

.container {
    box-shadow: 0 3px 6px rgba(0,0,0,0.16), 0 3px 6px
    rgba(0,0,0,0.23);
    transition: all 0.3s cubic-bezier(.25,.8,.25,1);
}
.container:hover {
    box-shadow: 0 18px 34px rgba(0,0,0,0.25), 0 20px 20px
    rgba(0,0,0,0.22);
}

.title {
    margin: 0;
    padding: 20px;
    background: dodgerblue;
    display: flex;
}

.title p{
    margin: 0;
    /* padding: 0; */
    font-size: 40px;
    color: white;
    flex-grow: 1;
}

#score {
    color: mediumpurple;
    text-align: center;
    font-size: 30px;
}

.container #question {
    /* font-family: "monospace"; */
    font-size: 30px;
    color: dodgerblue;
}

.buttons {
```

```

        margin-top: 30px;
    }

#btn0, #btn1, #btn2, #btn3 {
    background-color: dodgerblue;
    width: 250px;
    font-size: 20px;
    color: #fff;
    border: 1px solid dodgerblue;
    margin: 10px 40px 10px 0px;
    padding: 10px 10px;
}

#btn0:hover, #btn1:hover, #btn2:hover, #btn3:hover {
    cursor: pointer;
    background-color: dodgerblue;
}

#btn0:focus, #btn1:focus, #btn2:focus, #btn3:focus {
    outline: 0;
}

#progress {
    color: white;
    font-size: 18px;
}

footer{
    background: dodgerblue;
    padding: 5px 0;
    text-align: center;
    color: white;
    /* font-weight: bold; */
}

```

OUTPUT –

Let's Take a Quiz

Who among the following proposed Helio-centric model of solar system?

Copernicus
Ptolemy
Aristotle
Strabo

Question 1 of 10
 © 2020 Copyright, Quiz
 Made with ❤ by Bipin Lala

Let's Take a Quiz

Which of the following is the least dense planet among all the planets?

Earth

Uranus

Jupiter

Saturn

Question 10 of 10

© 2020 Copyright, Quiz

Made with ❤ by Bipin Lala

Result

Your scores: 8

PRACTICAL – 7B

AIM – Write a program to display 10 questions at random out of Practical-7A -10 questions

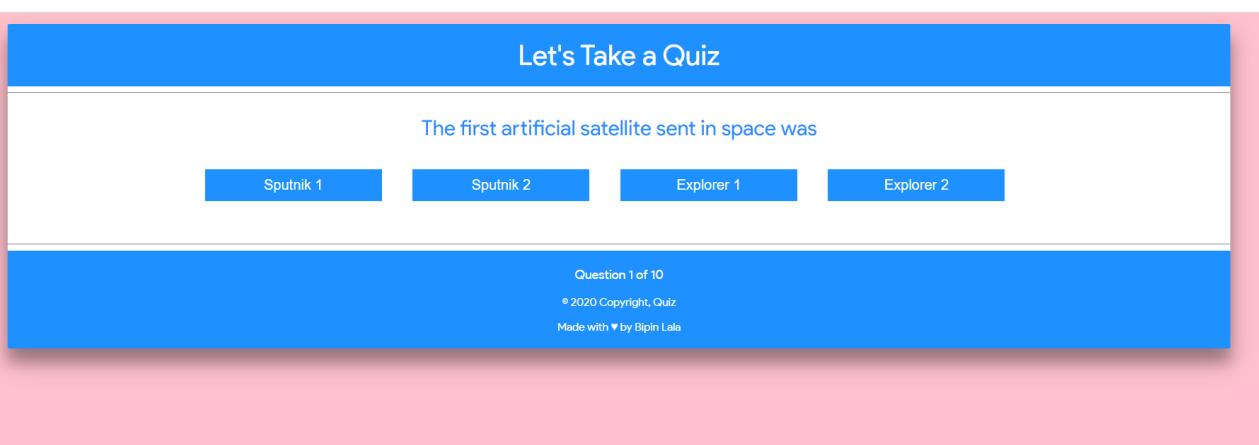
SOFTWARE USED – VS Code, Chrome

SOURCE CODE –

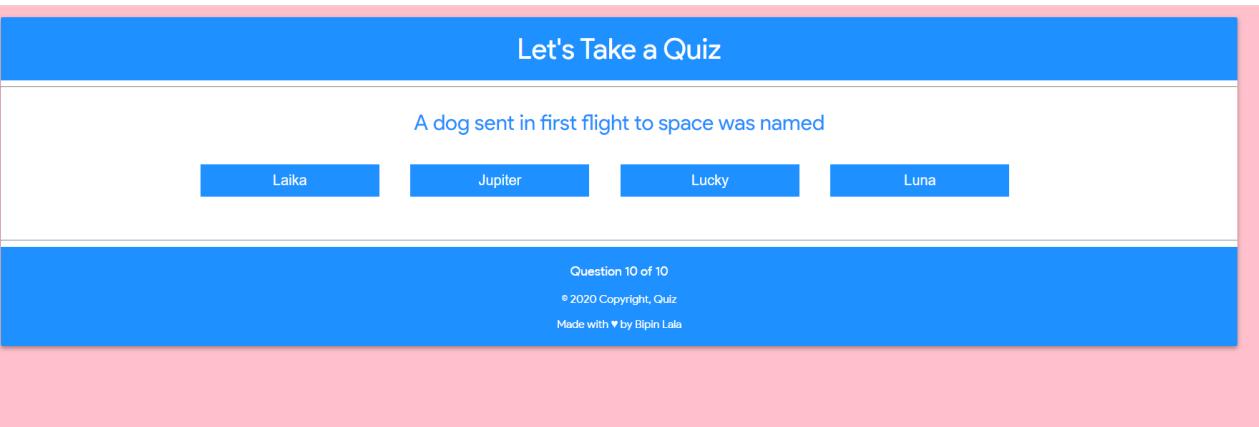
We need to write a small function that can randomize the questions. Since the questions are stored in a JavaScript array, we can randomize the array to shuffle the order of the questions.

```
// function to shuffle the questions  
  
function shuffle(array) {  
    array.sort(() => Math.random() - 0.5);  
}  
  
// calling function shuffle on array questions  
shuffle(questions)
```

OUTPUT –



The screenshot shows a mobile application interface for a quiz. At the top, a blue header bar displays the text "Let's Take a Quiz". Below this, the main content area has a white background. A question is displayed: "The first artificial satellite sent in space was". Four blue rectangular buttons are arranged horizontally below the question, labeled "Sputnik 1", "Sputnik 2", "Explorer 1", and "Explorer 2". At the bottom of the screen, a solid blue footer bar contains the text "Question 1 of 10", "© 2020 Copyright, Quiz", and "Made with ❤ by Bipin Lala".



The screenshot shows the same quiz application interface, but now it is displaying Question 10 of 10. The question is "A dog sent in first flight to space was named". The four options are "Laika", "Jupiter", "Lucky", and "Luna", each represented by a blue button. The layout is identical to the previous screenshot, with a blue header, white main content area, and a blue footer bar at the bottom containing copyright information.

PRACTICAL – 8

AIM – Implement Producer-Consumer Problem using Threads.

SOFTWARE USED – Eclipse

SOURCE CODE –

```
import java.util.LinkedList;

public class Threadexample {
    public static void main(String[] args)
        throws InterruptedException
    {
        // Object of a class that has both produce()
        // and consume() methods
        final PC pc = new PC();

        // Create producer thread
        Thread t1 = new Thread(new Runnable() {
            @Override
            public void run()
            {
                try {
                    pc.produce();
                }
                catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        });
        // Create consumer thread
        Thread t2 = new Thread(new Runnable() {
            @Override
            public void run()
            {
                try {
                    pc.consume();
                }
                catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        });
        // Start both threads
        t1.start();
        t2.start();

        // t1 finishes before t2
```

```

        t1.join();
        t2.join();
    }

    // This class has a list, producer (adds items to list
    // and consumer (removes items).
    public static class PC {

        // Create a list shared by producer and consumer
        // Size of list is 2.
        LinkedList<Integer> list = new LinkedList<>();
        int capacity = 2;

        // Function called by producer thread
        public void produce() throws InterruptedException
        {
            int value = 0;
            while (true) {
                synchronized (this)
                {
                    // producer thread waits while list
                    // is full
                    while (list.size() == capacity)
                        wait();

                    System.out.println("Producer
produced-"
+ value);

                    // to insert the jobs in the list
                    list.add(value++);

                    // notifies the consumer thread that
                    // now it can start consuming
                    notify();

                    // makes the working of program
easier
                    // to understand
                    Thread.sleep(1000);
                }
            }
        }

        // Function called by consumer thread
        public void consume() throws InterruptedException
        {
            while (true) {
                synchronized (this)
                {
                    // consumer thread waits while list
                    // is empty
                    while (list.size() == 0)
                        wait();

```

```
// to retrive the ifrst job in the
list
        int val = list.removeFirst();

        System.out.println("Consumer
consumed-"
                           + val);

        // Wake up producer thread
        notify();

        // and sleep
        Thread.sleep(1000);
    }
}
}
}
```

OUTPUT –

```
$javac Threadexample.java  
  
$java -Xmx128M -Xms16M Threadexample  
  
Producer produced-0  
Producer produced-1  
Consumer consumed-0  
Consumer consumed-1  
Producer produced-2  
Producer produced-3  
Consumer consumed-2  
Consumer consumed-3
```

EXCERCISE

QUES – Try to make your program produce one item and immediately after that make the consumer consume it before any other item is produced by the producer.

SOLUTION – We need to alter the code such that the consumer immediately consumes the item before another item is produced by the producer. Such behaviour can easily be reproduced by replacing

while (list.size() == capacity)

with

while (list.size() == 1) in the following code.

SOURCE CODE –

```
import java.util.LinkedList;

public class Threadexample {
    public static void main(String[] args)
        throws InterruptedException
    {
        // Object of a class that has both produce()
        // and consume() methods
        final PC pc = new PC();

        // Create producer thread
        Thread t1 = new Thread(new Runnable() {
            @Override
            public void run()
            {
                try {
                    pc.produce();
                }
                catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        });
        });

        // Create consumer thread
        Thread t2 = new Thread(new Runnable() {
            @Override
            public void run()
            {
                try {
                    pc.consume();
                }
            }
        });
    }
}
```

```

        catch (InterruptedException e) {
            e.printStackTrace();
        }
    });

// Start both threads
t1.start();
t2.start();

// t1 finishes before t2
t1.join();
t2.join();
}

// This class has a list, producer (adds items to list
// and consumer (removes items).
public static class PC {

    // Create a list shared by producer and consumer
    // Size of list is 2.
    LinkedList<Integer> list = new LinkedList<>();
    int capacity = 2;

    // Function called by producer thread
    public void produce() throws InterruptedException
    {
        int value = 0;
        while (true) {
            synchronized (this)
            {
                // producer thread waits while list
                // is full
                while (list.size() == 1)
                    wait();

                System.out.println("Producer
produced-"
                        + value);

                // to insert the jobs in the list
                list.add(value++);

                // notifies the consumer thread that
                // now it can start consuming
                notify();
            }

            // makes the working of program
            easier
            // to understand
            Thread.sleep(1000);
        }
    }
}

```

```

// Function called by consumer thread
public void consume() throws InterruptedException
{
    while (true) {
        synchronized (this)
        {
            // consumer thread waits while list
            // is empty
            while (list.size() == 0)
                wait();

            // to retrieve the first job in the
            list
            int val = list.removeFirst();

            System.out.println("Consumer
consumed-"
                               + val);

            // Wake up producer thread
            notify();

            // and sleep
            Thread.sleep(1000);
        }
    }
}

```

OUTPUT –

<pre> \$javac Threadexample.java \$java -Xmx128M -Xms16M Threadexample Producer produced-0 Consumer consumed-0 Producer produced-1 Consumer consumed-1 Producer produced-2 Consumer consumed-2 Producer produced-3 Consumer consumed-3 </pre>

CONTENT BEYOND SYLLABUS

Develop a console application for Movie Booking System including features- Book Ticket, Show My Ticket, Check Seat, Add new movie, Edit Movie, Delete Movie and more.

SOURCE CODE –

```
// Movie Reservation System
// Authors:
// Ujjwal Upadhyay - 03951202716
// Vipassana gautam - 04051202716
// Bipin Lala - 20151202716
// Aashita Arora - 20251202716

#include <iostream>
#include <fstream>
#include <string>
#include <stdio.h>
#include <time.h>

using namespace std;

int nCommand, nAccessLevel = 0;
string sUsername, sPassword;

void GuestMenu();
void Menu();
void Login();
void TicketPurchase();
void Movies();
void UpdateMovies(int i);
void AddMovies(int i);
void DeleteMovies(int i);
void UpdatePassword();
char CheckOccupancy(int CinemaHall, int Movie, int Show, int ChairNumber);
void ReserveChair(int CinemaHall, int Movie, int Show, int ChairNumber);
void ReserveChair(int CinemaHall, int Movie, int Show);
void SeatSelectionSystem(int nCinema, int nMovie, int nShowing);
void Reservation(int CinemaHall, int Movie, int Show, int Chair);
void DisplayHeader();
void reverse(char str[], int length)
{
    int start = 0;
    int end = length -1;
    while (start < end)
```

```

    {
        swap(* (str+start), * (str+end));
        start++;
        end--;
    }
}

char* itoa(int num, char* str, int base)
{
    int i = 0;
    bool isNegative = false;

    /* Handle 0 explicitely, otherwise empty string is printed
for 0 */
    if (num == 0)
    {
        str[i++] = '0';
        str[i] = '\0';
        return str;
    }

    // In standard itoa(), negative numbers are handled only
with
    // base 10. Otherwise numbers are considered unsigned.
    if (num < 0 && base == 10)
    {
        isNegative = true;
        num = -num;
    }

    // Process individual digits
    while (num != 0)
    {
        int rem = num % base;
        str[i++] = (rem > 9) ? (rem-10) + 'a' : rem + '0';
        num = num/base;
    }

    // If number is negative, append '-'
    if (isNegative)
        str[i++] = '-';

    str[i] = '\0'; // Append string terminator

    // Reverse the string
    reverse(str, i);

    return str;
}

std::string _strdate(char *cptime)
{
    time_t now = time(NULL);
    strftime(cptime, 50, "%b. %d, %Y", localtime(&now));
//short month name
    return std::string(cptime);
}

```

```

}

///////////////
// Main Function
/////////////
int main()

{
//    int a;
//    char c;
//    cin>>c;
//    a = c;
//    cout<<a;

Start:
DisplayHeader();
cout << " [1] Log In" << endl;
cout << " [2] Exit Application" << endl;
cout << endl;
cout << " Please select your comamnd: ";
cin >> nCommand;
switch (nCommand)
{
    case 1:
        Login();
        goto Start;
    case 2:
        return 0;
    default:
        goto Start;
}
//    return 0;
}

```

```

/////////////
// Login Function
/////////////
void Login()
{
    DisplayHeader();
    cout << " Enter your login details."<<endl;
    cout << " If logging in as a guest, type Guest as the
username."<<endl<<endl;
    cout << " Username: ";
    cin >> sUsername;

    // If username is Guest then log in as a guest
    if (sUsername == "Guest")
    {
        cout << " Logging in as a guest..." <<endl;
        cout << " ";
//        system("pause");
        GuestMenu();
    }
}
```

```

else if (sUsername == "admin") // If username is Admin
{
    bool bAccessAllowed = false;
    nAccessLevel = 2;
//    fstream FileHandler;
    cout << " Password: ";
    cin >> sPassword;
//    FileHandler.open("admin.txt");
    ifstream inFile;

inFile.open("/home/ujjwal/Documents/POPL/thesis/admin.txt");
    string pw;
    getline(inFile,pw); // Get one line and store it for
password
    if (sPassword == pw)
        bAccessAllowed = true;
//    FileHandler.close();
    if (bAccessAllowed == true)
    {
        cout << " Login successful." << endl;
        Menu();
    }
    else
    {
        cout << " Invalid username and/or password." <<
endl;
//        cout << " "; system("pause");
    }
}

else // If an ordinary user
{
    fstream FileHandler;
// If username is not Guest then ask for password
    bool bAccessAllowed = false;
    cout << " Password: ";
    cin >> sPassword;
    cout << endl;
// Open the userlist.upl file
    FileHandler.open("UserPass\\accounts.upl");
// While the file still has unread lines
    while ( FileHandler.good() )
    {
        string user;
        string pw;
        getline(FileHandler,user); // Get one line and
store it for username.
        getline(FileHandler,pw); // Get one line and store
it for password.
        if ((sUsername == user) && (sPassword == pw) )
        {
            bAccessAllowed = true;
            break;
        }
    }
}

```

```

        }
        // Closes the userlist.upl file
        FileHandler.close();
        if (bAccessAllowed == true)
        {
            cout << " Login successful." << endl;
        //         system("pause");
            Menu();
        }
        else
            cout << " Invalid username and/or password." <<
endl;
            cout << " ";
        }
    }

void GuestMenu()
{
    Start:
    DisplayHeader();
    cout << " [1] Buy ticket" << endl;
    cout << " [2] Verify reservation (Not done)" << endl;
    if (nAccessLevel == 0)
        cout << " [3] Log out" << endl;
    else
        cout << " [3] Back" << endl;
    cout << endl;
    cout << " Enter your command: ";
    cin >> nCommand;
    switch (nCommand)
    {
        case 1:
            TicketPurchase();
            break;
        case 3:
            return;
        default:
            break;
    }
    goto Start;
}

void Menu()
{
    Start:
    DisplayHeader();
    cout << " [1] Ticket Reservation" << endl;
    cout << " [2] Update Movies" << endl;
    cout << " [3] Change Password" << endl;
    cout << " [4] Log Out" << endl;
    cout << endl << " Please select your comamnd: ";
    cin >> nCommand;
    switch (nCommand)
    {

```

```

        case 1:
            GuestMenu();
            break;
        case 2:
            Movies();
            break;
        case 3:
            UpdatePassword();
        case 4:
            return;
        default:
            goto Start;
    }
    goto Start;
}

void TicketPurchase()
{
    SelectCinema:
    fstream Handler[2];
    int nCinema, nMovie, nShowing;
    DisplayHeader();
    cout << " Enter 0 to cancel." << endl;
    cout << " Enter cinema number [1-3]: ";
    cin >> nCinema;
    switch (nCinema)
    {
        case 1:
        case 2:
        case 3:
            break;
        case 0:
            return;
        default:
            goto SelectCinema;
    }
    SelectMovie:
    string sPrefix = "C*";
    char temp[2];
    itoa(nCinema,temp,10);
    sPrefix[1] = temp[0];
    string sFilename;
    char cFilename[18];
    int i;
    DisplayHeader();
    cout << " Enter 0 to cancel." << endl;
    // Movie 1
    sFilename = sPrefix + "\\M1\\Movie.props";
    for (i = 0; i < 18; i++)
    {
        cFilename[i] = sFilename[i];
    }
    Handler[0].open(cFilename);
    if (!Handler[0].fail())

```

```

{
    string sTitle;
    getline(Handler[0],sTitle);
    cout << " [1] " << sTitle << endl;
}
Handler[0].close();

// Movie 2
sFilename = sPrefix + "\M2\Movie.props";
for (i = 0; i < 18; i++)
{
    cFilename[i] = sFilename[i];
}
Handler[1].open(cFilename);
if (!Handler[1].fail())
{
    string sTitle;
    getline(Handler[1],sTitle);
    cout << " [2] " << sTitle << endl;
}
Handler[1].close();

cout << endl;
cout << " Enter movie number: ";
cin >> nMovie;

switch (nMovie)
{
    case 1:
        sFilename = sPrefix + "\M1\Movie.props";
        for (i = 0; i < 18; i++)
        {
            cFilename[i] = sFilename[i];
        }
        Handler[0].open(cFilename);
        if (!Handler[0].fail())
        {
            Handler[0].close();
            break;
        }
        else
        {
            Handler[0].close();
            goto SelectMovie;
        }
    case 2:
        sFilename = sPrefix + "\M2\Movie.props";
        for (i = 0; i < 18; i++)
        {
            cFilename[i] = sFilename[i];
        }
        Handler[1].open(cFilename);
        if (Handler[1].good())
        {

```

```

        Handler[1].close();
        break;
    }
    else
    {
        Handler[1].close();
        goto SelectMovie;
    }
    break;
case 0:
    return;
default:
    goto SelectMovie;
}
SelectShowing:
DisplayHeader();
cout << " Enter 0 to cancel." << endl;
cout << " Enter showing number [1-2]: ";
cin >> nShowing;

switch (nShowing)
{
    case 1:
    case 2:
        ReserveChair(nCinema, nMovie, nShowing);
        break;
    case 0:
        return;
    default:
        goto SelectShowing;
}
}

void Movies()
{
    Start:
    DisplayHeader();
    string title;
    ifstream Reader[6];
    cout << "[0] -Cancel-" << endl;

    Reader[0].open("C1\\M1\\Movie.props");
    if (Reader[0])
    {
        getline(Reader[0],title);
        cout << "[1] Cinema 1: " << title << endl;
    }
    else
        cout << "[1] Cinema 1: NO MOVIE" << endl;
    Reader[0].close();

    Reader[1].open("C1\\M2\\Movie.props");
    if (Reader[1])
    {

```

```

        getline(Reader[1],title);
        cout << " [2] Cinema 1: " << title << endl;
    }
    else
        cout << " [2] Cinema 1: NO MOVIE" << endl;
Reader[1].close();

Reader[2].open("C2\\M1\\Movie.props");
if (Reader[2])
{
    getline(Reader[2],title);
    cout << " [3] Cinema 2: " << title << endl;
}
else
    cout << " [3] Cinema 2: NO MOVIE" << endl;
Reader[2].close();

Reader[3].open("C2\\M2\\Movie.props");
if (Reader[3])
{
    getline(Reader[3],title);
    cout << " [4] Cinema 2: " << title << endl;
}
else
    cout << " [4] Cinema 2: NO MOVIE" << endl;
Reader[3].close();

Reader[4].open("C3\\M1\\Movie.props");
if (Reader[4])
{
    getline(Reader[4],title);
    cout << " [5] Cinema 3: " << title << endl;
}
else
    cout << " [5] Cinema 3: NO MOVIE" << endl;
Reader[4].close();

Reader[5].open("C3\\M2\\Movie.props");
if (Reader[5])
{
    getline(Reader[5],title);
    cout << " [6] Cinema 3: " << title << endl;
}
else
    cout << " [6] Cinema 3: NO MOVIE" << endl;
Reader[5].close();
cout << " Select which movie you will update: ";
cin >> nCommand;
switch (nCommand)
{
    case 1:
        UpdateMovies(nCommand);
        break;
    case 2:

```

```

        UpdateMovies(nCommand);
        break;
    case 3:
        UpdateMovies(nCommand);
        break;
    case 4:
        UpdateMovies(nCommand);
        break;
    case 5:
        UpdateMovies(nCommand);
        break;
    case 6:
        UpdateMovies(nCommand);
        break;
    case 0:
        return;
    default:
        goto Start;
    }
    goto Start;
}
void UpdateMovies(int i)
{
    Start:
    ifstream Reader;
    switch (i)
    {
        case 1:
            Reader.open("C1\\M1\\Movie.props");
            break;
        case 2:
            Reader.open("C1\\M2\\Movie.props");
            break;
        case 3:
            Reader.open("C2\\M1\\Movie.props");
            break;
        case 4:
            Reader.open("C2\\M2\\Movie.props");
            break;
        case 5:
            Reader.open("C3\\M1\\Movie.props");
            break;
        case 6:
            Reader.open("C3\\M2\\Movie.props");
            break;
    }
    DisplayHeader();
    if (Reader)
    {
        string title;
        getline(Reader,title);
        Reader.close();
        cout << " Movie title: " << title << endl;
        cout << " [1] Update Movie" << endl;
    }
}
```

```

cout << " [2] Delete Movie (Admin Only)" << endl;
cout << " [3] Back" << endl;
cout << endl << " Please enter your command: ";
int j;
cin >> j;
switch (j)
{
    case 1:
        AddMovies(i);
        return;
    case 2:
        if (nAccessLevel==2)
        {
            cout << "Movie has been successfully
deleted." << endl;
            DeleteMovies(i);
            return;
        }
        else
        {
            DisplayHeader();
            cout << " Function is only for admin."<<
endl;
            cout << " ";
            system("pause");
            goto Start;
        }
    case 3:
        return;
}
goto Start;
}
else
{
    cout << " [1] Add Movie" << endl;
    cout << " [2] Back" << endl;
    cout << endl;
    cout << " Please enter your command: ";
    int j;
    cin >> j;
    switch (j)
    {
        case 1:
            AddMovies(i);
            return;
        case 2:
            return;
        default:
            goto Start;
    }
    goto Start;
}
}

void AddMovies(int i)

```

```

{
    ofstream Writer;
    switch (i)
    {
        case 1:
            Writer.open("C1\\M1\\Movie.props");
            break;
        case 2:
            Writer.open("C1\\M2\\Movie.props");
            break;
        case 3:
            Writer.open("C2\\M1\\Movie.props");
            break;
        case 4:
            Writer.open("C2\\M2\\Movie.props");
            break;
        case 5:
            Writer.open("C3\\M1\\Movie.props");
            break;
        case 6:
            Writer.open("C3\\M2\\Movie.props");
            break;
        default:
            cout << "  ERROR: Invalid movie number." << endl;
//            cout << "  "; system("pause");
    }
    string Input;
    DisplayHeader();
    cout << "  ";
    getline(cin, Input); //Reset the cin;
//    Writer << Input << endl;
//    DisplayHeader();
    cout << "  Enter movie title: ";
    getline(cin, Input); //Reset the cin;
    Writer << Input << endl;
//    DisplayHeader();
    cout << "  Enter synopsis: ";
    getline(cin, Input);
    Writer << Input << endl;
//    DisplayHeader();
    cout << "  Enter price: ";
    getline(cin, Input);
    Writer << Input << endl;
//    DisplayHeader();
    cout << "  Enter date of showing [Month Day Year], ex. (Apr
01 2012): ";
    getline(cin, Input);
    Writer << Input << endl;
    Writer.close();
//    DisplayHeader();
    cout << "  Movie successfully updated!" << endl;
    cout << "  ";
//    system("pause");
}

```

```

void DeleteMovies(int i)
{
    char moviefile[]="C*\M*\Movie.props";
    char seatsfile[]="C*\M*\S*_***";
    switch (i)
    {
        case 1:
            moviefile[1]='1';
            seatsfile[1]='1';
            moviefile[4]='1';
            seatsfile[4]='1';
            break;
        case 2:
            moviefile[1]='1';
            seatsfile[1]='1';
            moviefile[4]='2';
            seatsfile[4]='2';
            break;
        case 3:
            moviefile[1]='2';
            seatsfile[1]='2';
            moviefile[4]='1';
            seatsfile[4]='1';
            break;
        case 4:
            moviefile[1]='2';
            seatsfile[1]='2';
            moviefile[4]='2';
            seatsfile[4]='2';
            break;
        case 5:
            moviefile[1]='3';
            seatsfile[1]='3';
            moviefile[4]='1';
            seatsfile[4]='1';
            break;
        case 6:
            moviefile[1]='3';
            seatsfile[1]='3';
            moviefile[4]='2';
            seatsfile[4]='2';
            break;
    }
    int res = remove(moviefile);
    for (i = 1; i <= 2; i++)
    {
        char temp[4];
        itoa(i,temp,10);
        seatsfile[7]=temp[0];
        for (int j = 1; j <= 250; j++)
        {
            if (j > 99)
            {
                itoa(j, temp, 10);

```

```

        seatsfile[9]=temp[0];
        seatsfile[10]=temp[1];
        seatsfile[11]=temp[2];
    }
    else if (j > 9)
    {
        itoa(j, temp, 10);
        seatsfile[9]='0';
        seatsfile[10]=temp[0];
        seatsfile[11]=temp[1];
    }
    else
    {
        itoa(j, temp, 10);
        seatsfile[9]='0';
        seatsfile[10]='0';
        seatsfile[11]=temp[0];
    }
    res = remove(seatsfile);

}
}

//      system("pause");
}

void UpdatePassword()
{
    DisplayHeader();
    string Input;
    string pw;
    cout << " Enter current password (Max 20 chars): ";
    getline(cin,Input);
    getline(cin,Input);
    pw = Input;
    if (!(pw==sPassword))
    {
        cout << " Wrong password!" << endl;
        return;
    }
    cout << " Enter new password (Max 20 chars): ";
    getline(cin,Input);
    pw = Input;
    string filename = "UserPass\\"+sUsername+"."+pw";
    const char* cFilename = filename.c_str();
    ofstream Writer;
    Writer.open(cFilename);
    Writer << pw;
    DisplayHeader();
    cout << " Password successfully changed." << endl;
//      cout << " "; system("pause");
}

char CheckOccupancy(int CinemaHall, int Movie, int Show, int
ChairNumber)

```

```

{
    ifstream Reader;
    char temp[4];
    char filename[]="C*\M*\S*_***";
    itoa(CinemaHall, temp, 10);
    filename[1]=temp[0];
    itoa(Movie, temp, 10);
    filename[4]=temp[0];
    itoa(Show, temp, 10);
    filename[7]=temp[0];
    itoa(ChairNumber, temp, 10);
    if (ChairNumber > 99)
    {
        filename[9]=temp[0];
        filename[10]=temp[1];
        filename[11]=temp[2];
    }
    else if (ChairNumber > 9)
    {
        filename[9]='0';
        filename[10]=temp[0];
        filename[11]=temp[1];
    }
    else
    {
        filename[9]='0';
        filename[10]='0';
        filename[11]=temp[0];
    }
    Reader.open(filename);
    if (!Reader)
        return 'O';
    else
        return 'X';
}

void ReserveChair(int CinemaHall, int Movie, int Show, int
ChairNumber)
{
    ofstream Writer;
    ifstream Reader;
    char temp[4];
    char filename[]="C*\M*\S*_***";
    char fileopen[]="C*\M*\Movie.props";
    itoa(CinemaHall, temp, 10);
    filename[1]=temp[0];
    fileopen[1]=temp[0];
    itoa(Movie, temp, 10);
    filename[4]=temp[0];
    fileopen[4]=temp[0];
    itoa(Show, temp, 10);
    filename[7]=temp[0];
    itoa(ChairNumber, temp, 10);
    if (ChairNumber > 99)

```

```

{
    filename[9]=temp[0];
    filename[10]=temp[1];
    filename[11]=temp[2];
}
else if (ChairNumber > 9)
{
    filename[9]='0';
    filename[10]=temp[0];
    filename[11]=temp[1];
}
else
{
    filename[9]='0';
    filename[10]='0';
    filename[11]=temp[0];
}
Writer.open(filename);
// First * - Random Letter
// Next 2 * - Reservation Day
// Next * - Random Letter
// Next 2 * - Reservation Month
// Next * - Random Letter
// Next 2 * - Reservation Year
// Next * - Cinema Number
// Next * - Movie Number
// Next 2 * - Showing (AM/PM)
// Next 3 * - ChairNumber
char ticketn[]="*****C*M**S***";
char rletters[3];
char date[9];
int n;
srand(time(NULL));
n = rand() % 26;
rletters[0] = (char)(n+65);
n = rand() % 26;
rletters[1] = (char)(n+65);
n = rand() % 26;
rletters[2] = (char)(n+65);
_strdate(date);
ticketn[0] = rletters[0];
ticketn[1] = date[0];
ticketn[2] = date[1];
ticketn[3] = rletters[1];
ticketn[4] = date[3];
ticketn[5] = date[4];
ticketn[6] = rletters[2];
ticketn[7] = date[6];
ticketn[8] = date[7];
itoa(CinemaHall, temp, 10);
ticketn[10] = temp[0];
itoa(Movie, temp, 10);
ticketn[12] = temp[0];
if (Show == 1)

```

```

        ticketn[13]='A';
    else
        ticketn[13]='P';
ticketn[14]='M';
itoa(ChairNumber, temp, 10);
if (ChairNumber > 99)
{
    ticketn[16]=temp[0];
    ticketn[17]=temp[1];
    ticketn[18]=temp[2];
}
else if (ChairNumber > 9)
{
    ticketn[16]='0';
    ticketn[17]=temp[0];
    ticketn[18]=temp[1];
}
else
{
    ticketn[16]='0';
    ticketn[17]='0';
    ticketn[18]=temp[0];
}
DisplayHeader();
Reader.open(fileopen);
string sTitle;
getline(Reader,sTitle);
time_t curr = time(0);
cout <<"\tMovie Title: " << sTitle << endl;
cout << "\tCinema: " << CinemaHall << "\t\tShowing: ";
if (Show == 1)
    cout <<"AM";
else
    cout <<"PM";
cout<<endl;
cout <<"\tTime of Purchase:\t\t" << ctime(&curr) << endl;
cout << "    Ticket ID: ";
for (int i = 0; i < sizeof(ticketn) - 1; i++)
{
    cout << ticketn[i];
    Writer << ticketn[i];
}
Writer << "\n" << ctime(&curr);
Writer.close();
cout << "\tBe sure to write down your ticket ID!"<<endl;
//    system("pause");
}

void ReserveChair(int CinemaHall, int Movie, int Show)
{
    ofstream FileWriter;
    ifstream FileChecker;
    char cFilename[15];
    char temp[2];

```

```

string sFilename = "C*\M*\S*.seats";
itoa(CinemaHall, temp, 10);
sFilename[1] = temp[0];
itoa(Movie, temp, 10);
sFilename[4] = temp[0];
itoa(Show, temp, 10);
sFilename[7] = temp[0];
for (int j = 0; j < 15; j++)
{
    cFilename[j] = sFilename[j];
}
FileChecker.open(cFilename);
FileChecker.close();
SeatSelectionSystem(CinemaHall, Movie, Show);
}

void SeatSelectionSystem(int Cinema, int Movie, int Show)
{
    int nCursorX = 1, nCursorY = 0;
Start:
//    system("clear");
cout << endl << endl;
if ((nCursorX>=1) && (nCursorX<=10))
{
    cout << "      Area 1:" << endl;
    for (int i = 0; i <= 200; i+=50)
    {
        cout << "      ";
        for (int j = 1; j <= 10; j++)
        {
            if (nCursorX+(nCursorY*50)==i+j)
                cout << " <<" << CheckOccupancy(Cinema,
Movie, Show, i + j) << ">> ";
            else
                cout << "  [ " << CheckOccupancy(Cinema,
Movie, Show, i + j) << " ]  ";
        }
        cout << endl << endl;
    }
}
else if ((nCursorX>=11) && (nCursorX<=20))
{
    cout << "      Area 2:" << endl;
    for (int i = 0; i <= 200; i+=50)
    {
        cout << "      ";
        for (int j = 11; j <= 20; j++)
        {
            if (nCursorX+(nCursorY*50)==i+j)
                cout << " <<" << CheckOccupancy(Cinema,
Movie, Show, i + j) << ">> ";
            else
                cout << "  [ " << CheckOccupancy(Cinema,
Movie, Show, i + j) << " ]  ";
        }
    }
}

```

```

        }
        cout << endl << endl;
    }
}
else if ((nCursorX>=21)&&(nCursorX<=30))
{
    cout << "      Area 3:" << endl;
    for (int i = 0; i <= 200; i+=50)
    {
        cout << "      ";
        for (int j = 21; j <= 30; j++)
        {
            if (nCursorX+(nCursorY*50)==i+j)
                cout << " <<" << CheckOccupancy(Cinema,
Movie, Show, i + j) << ">> ";
            else
                cout << "  [ " << CheckOccupancy(Cinema,
Movie, Show, i + j) << " ]  ";
        }
        cout << endl << endl;
    }
}
else if ((nCursorX>=31)&&(nCursorX<=40))
{
    cout << "      Area 4:" << endl;
    for (int i = 0; i <= 200; i+=50)
    {
        cout << "      ";
        for (int j = 31; j <= 40; j++)
        {
            if (nCursorX+(nCursorY*50)==i+j)
                cout << " <<" << CheckOccupancy(Cinema,
Movie, Show, i + j) << ">> ";
            else
                cout << "  [ " << CheckOccupancy(Cinema,
Movie, Show, i + j) << " ]  ";
        }
        cout << endl << endl;
    }
}
else
{
    cout << "      Area 5:" << endl;
    for (int i = 0; i <= 200; i+=50)
    {
        cout << "      ";
        for (int j = 41; j <= 50; j++)
        {
            if (nCursorX+(nCursorY*50)==i+j)
                cout << " <<" << CheckOccupancy(Cinema,
Movie, Show, i + j) << ">> ";
            else
                cout << "  [ " << CheckOccupancy(Cinema,
Movie, Show, i + j) << " ]  ";
    }
}

```

```

        }
        cout << endl << endl;
    }
}

cout << "      Legend:<<endl;
cout << "          O -> Available" << endl;
cout << "          X -> Unavailable" << endl << endl << endl;
cout << "\tUse the Arrow Keys to move, Space to select and
Esc for cancel.";

char c_z;
cin>>c_z;
int z = c_z;
switch (z)
{
    case 115:
        if (nCursorY<4)
            nCursorY++;
        break;
    case 119:
        if (nCursorY>0)
            nCursorY--;
        break;
    case 97:
        if (nCursorX>1)
            nCursorX--;
        break;
    case 100:
        if (nCursorX<50)
            nCursorX++;
        break;
    case 120:
        char cCheck;
        cCheck = CheckOccupancy(Cinema, Movie, Show,
nCursorX + (nCursorY * 50));
        if (cCheck=='O')
            Reservation(Cinema, Movie, Show, nCursorX +
(nCursorY * 50));
        goto Start;
    case 122:
        return;
    default:
        break;
}
goto Start;
}

void Reservation(int CinemaHall, int Movie, int Show, int
Chair)
{
    char x;
    cout << "\t\t\t\t\t\tReserve this seat[Y/N] ? ";
    cin >> x;
    if (tolower(x)=='y')
        ReserveChair(CinemaHall, Movie, Show, Chair);
}

```

```

}

void DisplayHeader()
{
//    system("clear");
    cout << endl;

    // Upper Border
    cout << " \xC9";
    for (int i = 0; i < 76; i++)
        cout << "\xCD";
    cout << "\xBB";
    cout << endl;

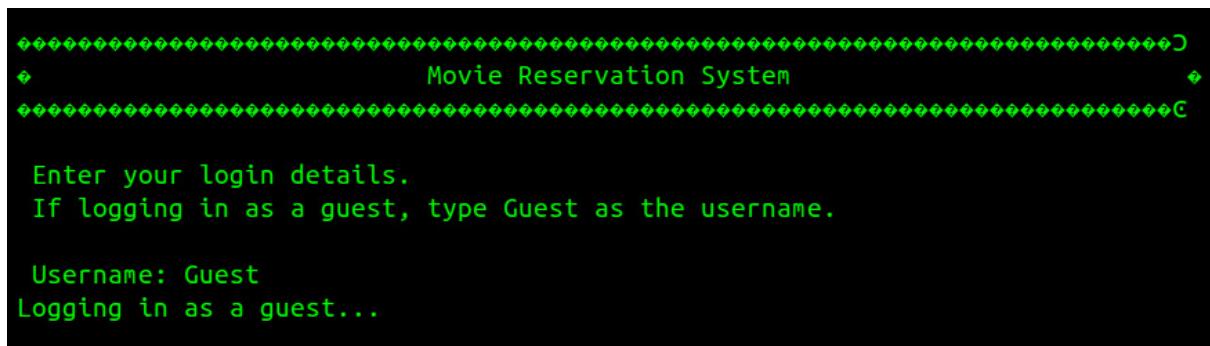
    // Middle Border
    cout << " \xBA";
    //      [FROM_HERE
TO_HERE]
    cout << "                               Movie Reservation
System               ";
    cout << "\xBA";
    cout << endl;

    // Upper Border
    cout << " \xC8";
    for (int i = 0; i < 76; i++)
        cout << "\xCD";
    cout << "\xBC";
    cout << endl;
    cout << endl;
}

```

OUTPUT –

Guest Login:



The terminal window displays the following output:

```

***** Movie Reservation System *****

Enter your login details.
If logging in as a guest, type Guest as the username.

Username: Guest
Logging in as a guest...

```

Admin Login:

```
***** Movie Reservation System *****  
* [1] Log In  
* [2] Exit Application  
*  
Please select your comamnd: 1  
  
***** Movie Reservation System *****  
* [1] Log In  
* [2] Exit Application  
*  
Enter your login details.  
If logging in as a guest, type Guest as the username.  
  
Username: admin  
Password: test  
Login successful.
```

Password Update for Admin:

```
***** Movie Reservation System *****  
* [1] Ticket Reservation  
* [2] Update Movies  
* [3] Change Password  
* [4] Log Out  
*  
Please select your comamnd: 3  
  
***** Movie Reservation System *****  
* [1] Ticket Reservation  
* [2] Update Movies  
* [3] Change Password  
* [4] Log Out  
*  
Enter current password (Max 20 chars): test  
Enter new password (Max 20 chars): test1  
  
***** Movie Reservation System *****  
* [1] Ticket Reservation  
* [2] Update Movies  
* [3] Change Password  
* [4] Log Out  
*  
Password successfully changed.
```

Exit Screen:

```
***** Movie Reservation System *****  
* [1] Log In  
* [2] Exit Application  
  
Please select your command: 2  
ujjwal@Newton ~/Documents/POPL/thesis ➤
```

Movie Addition:

```
***** Movie Reservation System *****  
* [1] Add Movie  
* [2] Back  
  
Please enter your command: 1  
  
***** Movie Reservation System *****  
* Enter movie title: Ujjwal visits madagaskar  
Enter synopsis: ujjwal visits madagaskar but due to love and affection from people over there he decides to open a new branch of BVP there and during this effort he was elected the president of madagaskar. See what it brings for India.  
Enter price: 3000  
Enter date of showing [Month Day Year], ex.(Apr 01 2012): Jun 02 2022  
Movie successfully updated!
```

Movie Update:

```
***** Movie Reservation System *****  
Movie title: Ujjwal visits madagaskar  
[1] Update Movie  
[2] Delete Movie (Admin Only)  
[3] Back  
  
Please enter your command: 1  
  
***** Movie Reservation System *****  
Enter movie title: Ujjwal and code refactoring  
Enter synopsis: ujjwal refactors code just before submission deadline  
Enter price: 4000  
Enter date of showing [Month Day Year], ex.(Apr 01 2012): May 03 2020  
Movie successfully updated!
```

Movie Delete:

```
***** Movie Reservation System *****  
[0] -Cancel-  
[1] Cinema 1: Vipassana & Oil Economy  
[2] Cinema 1: Ujjwal and code refactoring  
[3] Cinema 2: Bipin - The commandar in chief  
[4] Cinema 2: NO MOVIE  
[5] Cinema 3: NO MOVIE  
[6] Cinema 3: NO MOVIE  
Select which movie you will update: 3  
  
***** Movie Reservation System *****  
Movie title: Bipin - The commandar in chief  
[1] Update Movie  
[2] Delete Movie (Admin Only)  
[3] Back  
  
Please enter your command: 2  
Movie has been successfully deleted.
```

Ticket Confirmation:

```
Reserve this seat[Y/N]? y

***** Movie Reservation System *****

Movie Title: Vipassana & Oil Economy
Cinema: 1 Showing: AM
Time of Purchase: Sat May 2 20:55:57 2020

Ticket ID: WMaQ. E2,C1M1AMS051 Be sure to write down your ticket ID!
```

Ticket Reservation-1:

```
***** Movie Reservation System *****

[1] Ticket Reservation
[2] Update Movies
[3] Change Password
[4] Log Out

Please select your comamnd: 1

***** Movie Reservation System *****

[1] Buy ticket
[2] Verify reservation (Not done)
[3] Back

Enter your command: 1

***** Movie Reservation System *****

Enter 0 to cancel.
Enter cinema number [1-3]: 1

***** Movie Reservation System *****

Enter 0 to cancel.
[1] Vipassana & Oil Economy
[2] Ujjwal and code refactoring

Enter movie number: 1
```

Movie Reservation-2:

```
*****  
* Movie Reservation System *  
*****  
  
Enter 0 to cancel.  
Enter showing number [1-2]: 1  
  
Area 1:  
<<0>> [0] [0] [0] [0] [0] [0] [0] [0] [0]  
[0] [0] [0] [0] [0] [0] [0] [0] [0]  
[0] [0] [0] [0] [0] [0] [0] [0] [0]  
[0] [0] [0] [0] [0] [0] [0] [0] [0]  
[0] [0] [0] [0] [0] [0] [0] [0] [0]  
  
Legend:  
0 -> Available  
X -> Unavailable  
  
Use the Arrow Keys to move, Space to select and Esc for cancel.s  
  
Area 1:  
[0] [0] [0] [0] [0] [0] [0] [0] [0]  
<<0>> [0] [0] [0] [0] [0] [0] [0] [0] [0]  
[0] [0] [0] [0] [0] [0] [0] [0] [0]  
[0] [0] [0] [0] [0] [0] [0] [0] [0]  
[0] [0] [0] [0] [0] [0] [0] [0] [0]
```