Rayat Shikshan Sanstha's

KARMAVEER BHAURAO PATIL COLLEGE, VASHI

[Autonomous College]

Reaccredited by NAAC with Grade 'A+' (CGPA 3.53) | ISO 9001: 2008 Certified Institute

'Best College' Award by University of Mumbai

Programming Paradigm

Name: - Vinit Rajendra Ghodekar.

Class: - MSC-CS I

Roll No: - 240524

Subject: - Programing Paradigm

INDEX

SR.NO	PRACTICAL	DATE	SIGN
1	Implement the Bubble Sort algorithm	26-07-2024	
2	Implement operations on a linked list (e.g., insertion, deletion).	09-08-2024	
3	a) Write a program that calculates grades based on marks using selection (if-else) and iteration (loop) structures. b) Write modular functions to convert temperature between Celsius and Fahrenheit.	16-08-2024	
4	a) Write functions to traverse and print elements of an array using iteration. b) Implement a dynamic programming solution using arrays, such as the Fibonacci sequence or the Knapsack problem.	30-08-2024	
5	Write functions to dynamically allocate memory for arrays using pointers, and handle memory deallocation.	20-09-2024	
6	a) Write functions using map, filter, and reduce to transform, filter, and aggregate data respectively. b) Implement operations on immutable data structures (e.g., tuples, frozen sets) using functional programming principles.	27-09-2024	
7	To develop a real-time chat application using event-driven programming principles, ensuring that messages are sent and received instantly between clients without refreshing the page.	04-10-2024	
8	Implement using Go Programming : a) Calculate sum of first n numbers. b) Recursive function to find factorial of a number.	11-10-2024	
9	Implement using Ruby programming : a) Define classes and create objects with attributes and methods. b) Implement inheritance and demonstrate polymorphic behavior.	18-10-2024	
10.	Implement a functional program to calculate the sum of squares of even numbers from a list using Haskell.	25-40-2024	

Practical No 1

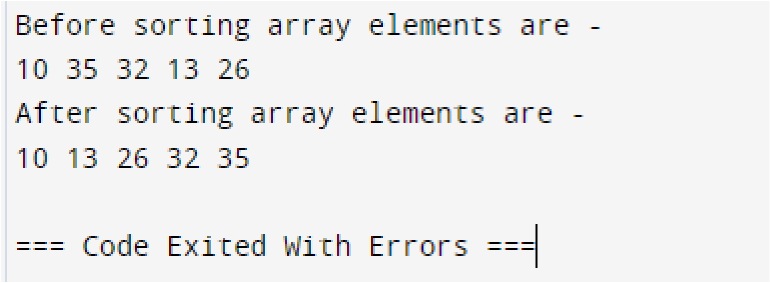
AIM: Implement the Bubble Sort algorithm

CODE:

A] c language

```
#include<stdio.h>
 void print(int a[], int n) //function to print array elements
    {
    int i;
    for(i = 0; i < n; i++)
    {
        printf("%d ",a[i]);
    }
    }
 void bubble(int a[], int n) // function to implement bubble sort
 {
   int i, j, temp;
   for(i = 0; i < n; i++)
    {
      for(j = i+1; j < n; j++)
        {
            if(a[j] < a[i])
            {
                temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
        }
    }
 }
void main ()
{
    int i, j,temp;
    int a[5] = { 10, 35, 32, 13, 26};
    int n = sizeof(a)/sizeof(a[0]);
    printf("Before sorting array elements are - \n");
    print(a, n);
    bubble(a, n);
    printf("\nAfter sorting array elements are - \n");
    print(a, n);
}
```

OUTPUT:



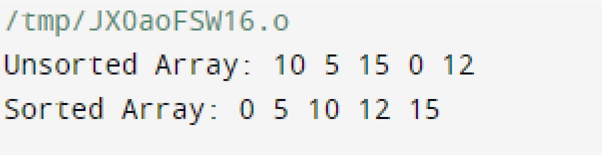
```
Before sorting array elements are -
10 35 32 13 26
After sorting array elements are -
10 13 26 32 35

=== Code Exited With Errors ===
```

B] C++

CODE:

```
#include <iostream>

void bubbleSort(int arr[], int n) {
    int i, j;
    bool flag;
    for(i = 0; i < n; i++) {
        flag = false;
        for(j = 0; j < n-i-1; j++) {
            if(arr[j] > arr[j+1]) {
                std::swap(arr[j], arr[j+1]);
                flag = true;
            }
        }
        if(!flag) {
            break;
        }
    }
}

int main() {
    int arr[] = {10, 5, 15, 0, 12};
    int n = sizeof(arr) / sizeof(arr[0]);

    std::cout << "Unsorted Array: ";
    for(int i = 0; i < n; i++)
        std::cout << arr[i] << " ";
    std::cout << std::endl;

    bubbleSort(arr, n);

    std::cout << "Sorted Array: ";
    for(int i = 0; i < n; i++)
        std::cout << arr[i] << " ";

    return 0;
}
```

OUTPUT:



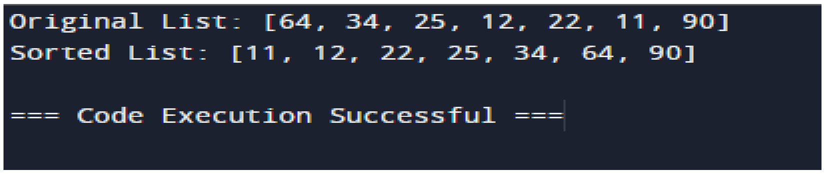
```
/tmp/JX0aoFSW16.o
Unsorted Array: 10 5 15 0 12
Sorted Array: 0 5 10 12 15
```

C]Python

CODE:

```
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
if _name___== "_main_":
    sample_list = [64, 34, 25, 12, 22, 11, 90]
    print("Original List:", sample_list)
    bubble_sort(sample_list)
    print("Sorted List:", sample_list)
```

OUTPUT:



```
Original List: [64, 34, 25, 12, 22, 11, 90]
Sorted List: [11, 12, 22, 25, 34, 64, 90]

=== Code Execution Successful ===
```

Practical No 2

AIM: Implement operations on a linked list (e.g., insertion, deletion).

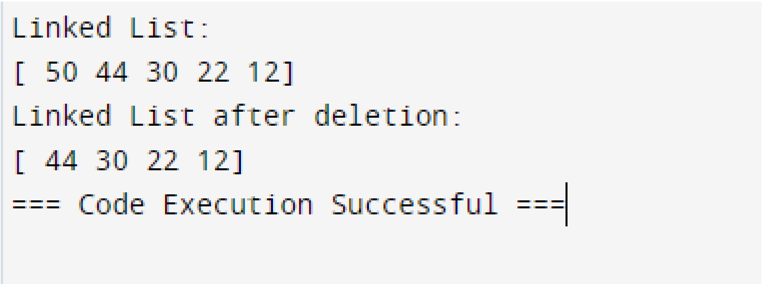
i) Insertion at Beginning in linked list.

CODE:

Insertion.java

```
public class Linked_List {

    static class Node {
        int data;
        Node next;

        Node(int value) {
            data = value;
            next = null;
        }
    }

    static Node head;

    static void printList() {
        Node p = head;
        System.out.print("\n[");
        while (p != null) {
            System.out.print(" " + p.data);
            p = p.next;
        }
        System.out.print("]");
    }

    static void insertAtBegin(int data) {
        Node newNode = new Node(data);
        newNode.next = head;
        head = newNode;
    }

    static void deleteAtBegin() {
        if (head != null) {
            head = head.next;
        }
    }
    public static void main(String[] args) {
        insertAtBegin(12);
        insertAtBegin(22);
        insertAtBegin(30);
        insertAtBegin(44);
        insertAtBegin(50);

        System.out.print("Linked List: ");
        printList();

        deleteAtBegin();

        System.out.print("\nLinked List after deletion: ");
        printList();
    }
}
```

OUTPUT:



```
Linked List:
[ 50 44 30 22 12]
Linked List after deletion:
[ 44 30 22 12]
=== Code Execution Successful ===
```

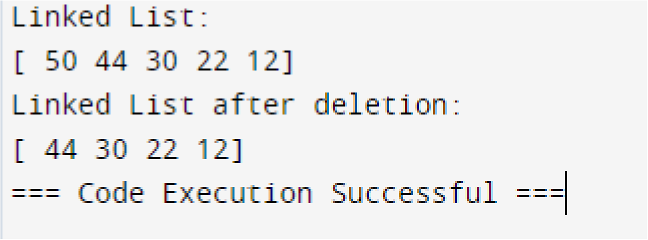
ii) Deletion at Beginning in linked list

CODE:

Deletion.java

```
public class Linked_List {

    static class Node {
        int data;
        Node next;

        Node(int value) {
            data = value;
            next = null;
        }
    }

    static Node head;

    static void printList() {
        Node p = head;
        System.out.print("\n[");
        while (p != null) {
            System.out.print(" " + p.data);
            p = p.next;
        }
        System.out.print("]");
    }

    static void insertAtBegin(int data) {
        Node newNode = new Node(data);
        newNode.next = head;
        head = newNode;
    }

    static void deleteAtBegin() {
        if (head != null) {
            head = head.next;
        }
    }

    public static void main(String[] args) {
        insertAtBegin(12);
        insertAtBegin(22);
        insertAtBegin(30);
        insertAtBegin(44);
        insertAtBegin(50);

        System.out.print("Linked List: ");
        printList();

        deleteAtBegin();

        System.out.print("\nLinked List after deletion: ");
        printList();
    }
}
```

OUTPUT:



```
Linked List:
[ 50 44 30 22 12]
Linked List after deletion:
[ 44 30 22 12]
=== Code Execution Successful ===
```

iii) Insertion using python

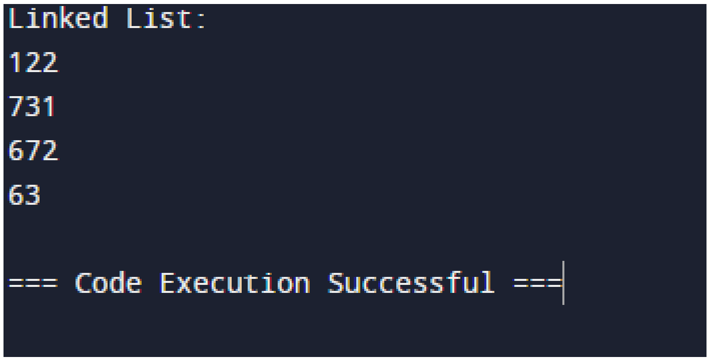
CODE:

Insertion.py

```
class Node:
    def __init__(self, data=None):
        self.data = data
        self.next = None

class SLL:
    def __init__(self):
        self.head = None

    # Print the linked list
    def listprint(self):
        printval = self.head
        print("Linked List: ")
        while printval is not None:
            print(printval.data)
            printval = printval.next

    def AddAtBeginning(self, newdata):
        newNode = Node(newdata)
        # Update the new node's next to the existing head
        newNode.next = self.head
        self.head = newNode

# Create the linked list and add nodes
l1 = SLL()
l1.head = Node("731")
e2 = Node("672")
e3  =  Node("63")
l1.head.next = e2
e2.next = e3
```

```
# Add a new node at the beginning
l1.AddAtBeginning("122")

# Print the linked list
l1.listprint()
```

OUTPUT:



```
Linked List:
122
731
672
63

=== Code Execution Successful ===
```

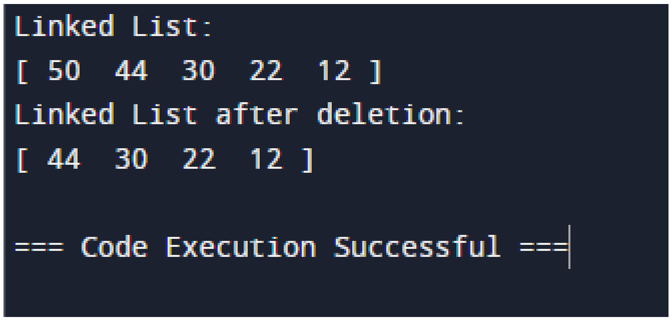
iv) Deletion using python

CODE:

Deletion.py

```
from typing import Optional

class Node:
    def __init__(self, data: int, next: Optional['Node'] = None):
        self.data = data
        self.next = next

class LinkedList:
    def __init__(self):
        self.head = None

    # Display the list
    def print_list(self):
        p = self.head
        print("\n[", end="")
        while p:
            print(f" {p.data} ", end="")
            p = p.next
        print("]")

    # Insert at the beginning
    def insert_at_begin(self, data: int):
        new_node = Node(data)
        # Point new node to the old first node
        new_node.next = self.head
        # Point head to the new first node
        self.head = new_node

    # Delete at the beginning
    def delete_at_begin(self):
        if self.head is not None:
            self.head = self.head.next

if _name___== "_main_":
    linked_list = LinkedList()
    linked_list.insert_at_begin(12)
    linked_list.insert_at_begin(22)
    linked_list.insert_at_begin(30)
    linked_list.insert_at_begin(44)
    linked_list.insert_at_begin(50)

    # Print list
    print("Linked List: ", end="")
    linked_list.print_list()

    linked_list.delete_at_begin()
    print("Linked List after deletion: ", end="")
    linked_list.print_list()
```

OUTPUT:



```
Linked List:
[ 50  44  30  22  12 ]
Linked List after deletion:
[ 44  30  22  12 ]

=== Code Execution Successful ===
```

Practical No 3

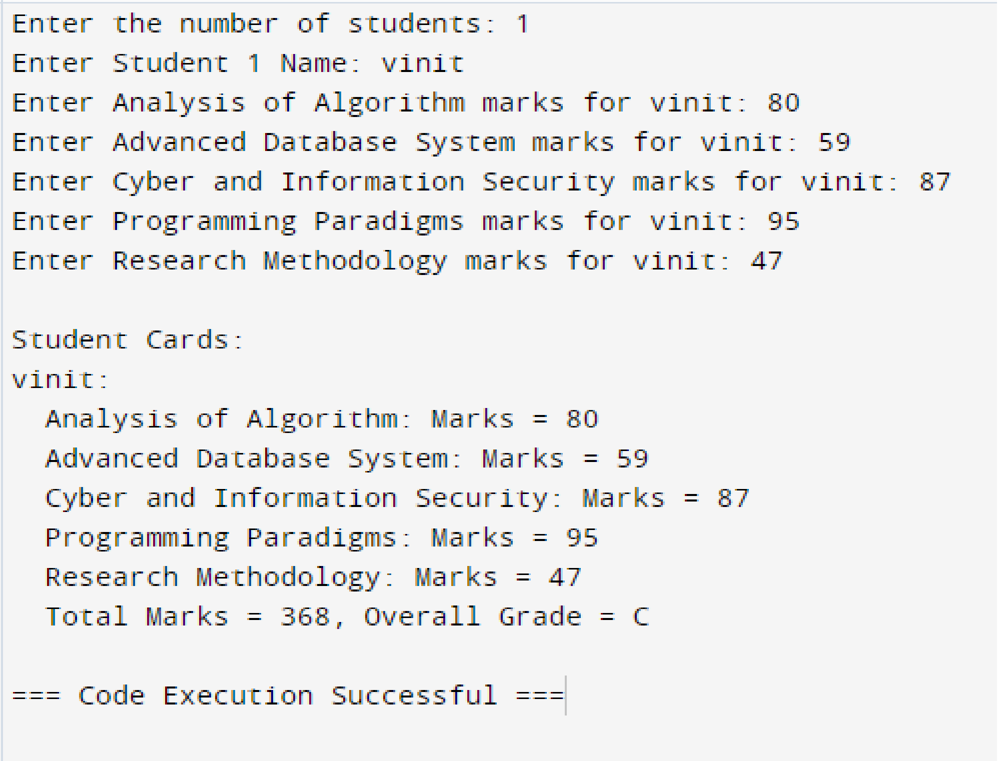
AIM: A)Write a program that calculates grades based on marks using selection (if-else) and iteration (loop) structures.

CODE:

python

```
def calculate_grade(marks):
    percentage=(marks/500)*100
    if 0 <= percentage <= 100:
        if 90 <= percentage <= 100:
            return 'A'
        elif 80 <= percentage < 90:
            return 'B'
        elif 70 <= percentage < 80:
            return 'C'
        elif 60 <= percentage < 70:
            return 'D'
        elif 35 <= percentage < 60:
            return 'E'
        else:
            return 'F'
    else:
        return 'Invalid'

def main():
    num_students = int(input("Enter the number of students: "))
    subjects = ["Analysis of Algorithm", "Advanced Database System", "Cyber and Information Security", "Programming Paradigms", "Research Methodology"]

    student_data = []
    for i in range(num_students):
        student_name = input(f"Enter Student {i + 1} Name: ")
        student_marks = []
        for subject in subjects:
            marks = int(input(f"Enter {subject} marks for {student_name}: "))
            student_marks.append(marks)
        student_data.append([student_name, student_marks])
    # Calculate total marks and overall grades for each student
    for student in student_data:
        name, marks = student
        total_marks = sum(marks)
        overall_grade = calculate_grade(total_marks)
        student.append(total_marks)
        student.append(overall_grade)

    # Print student cards
    print("\nStudent Cards:")
    for student in student_data:
        name, marks, total_marks, overall_grade = student
        print(f"{name}:")
        for i, subject in enumerate(subjects):
            print(f"  {subject}: Marks = {marks[i]}")
        print(f"  Total Marks = {total_marks}, Overall Grade = {overall_grade}")

if _name___== "_main_":
    main()
```

OUTPUT:



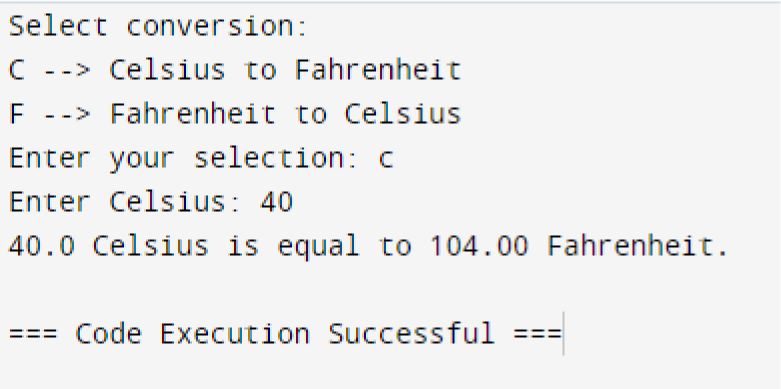
```
Enter the number of students: 1
Enter Student 1 Name: vinit
Enter Analysis of Algorithm marks for vinit: 80
Enter Advanced Database System marks for vinit: 59
Enter Cyber and Information Security marks for vinit: 87
Enter Programming Paradigms marks for vinit: 95
Enter Research Methodology marks for vinit: 47

Student Cards:
vinit:
  Analysis of Algorithm: Marks = 80
  Advanced Database System: Marks = 59
  Cyber and Information Security: Marks = 87
  Programming Paradigms: Marks = 95
  Research Methodology: Marks = 47
  Total Marks = 368, Overall Grade = C

=== Code Execution Successful ===
```

B)Write modular functions to convert temperature between Celsius and Fahrenheit.

CODE:

```
def celsius_to_fahrenheit(celsius):
    return (celsius * 1.8) + 32

def fahrenheit_to_celsius(fahrenheit):
    return (fahrenheit - 32) / 1.8

print("Select conversion:")
print("C --> Celsius to Fahrenheit")
print("F --> Fahrenheit to Celsius")

user_selection = input("Enter your selection: ").upper()  # Convert input to uppercase

if user_selection == "C":
    celsius = float(input("Enter Celsius: "))
    fahrenheit = celsius_to_fahrenheit(celsius)
    print(f"{celsius} Celsius is equal to {fahrenheit:.2f} Fahrenheit.")
elif user_selection == "F":
    fahrenheit = float(input("Enter Fahrenheit: "))
    celsius = fahrenheit_to_celsius(fahrenheit)
    print(f"{fahrenheit} Fahrenheit is equal to {celsius:.2f} Celsius.")
else:
    print("Enter a valid input (C or F).")
```

OUTPUT:



```
Select conversion:
C --> Celsius to Fahrenheit
F --> Fahrenheit to Celsius
Enter your selection: c
Enter Celsius: 40
40.0 Celsius is equal to 104.00 Fahrenheit.

=== Code Execution Successful ===
```

Practical No 4

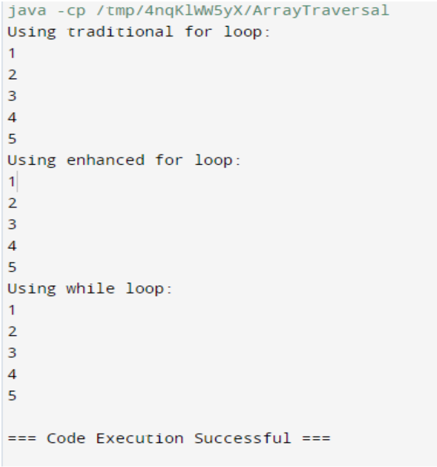
AIM: Write functions to traverse and print elements of an array using iteration.

CODE:

java

```
public class ArrayTraversal {
    public static void main(String[] args) {
        int[] numbers = {1, 2, 3, 4, 5};
        System.out.println("Using traditional for loop:");
        for (int i = 0; i < numbers.length; i++) {
            System.out.println(numbers[i]);
        }
        System.out.println("Using enhanced for loop:");
        for (int num : numbers) {
            System.out.println(num);
        }
        System.out.println("Using while loop:");
        int index = 0;
        while (index < numbers.length) {
            System.out.println(numbers[index]);
            index++;
        }
    }
}
```

OUTPUT:



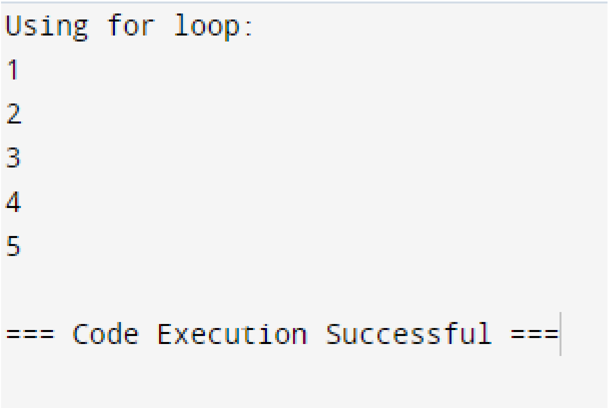
```
java -cp /tmp/4nqKlWW5yX/ArrayTraversal
Using traditional for loop:
1
2
3
4
5
Using enhanced for loop:
1
2
3
4
5
Using while loop:
1
2
3
4
5

=== Code Execution Successful ===
```

CODE:

python

```
def print_elements(arr):
    print("Using for loop:")
    for element in arr:
        print(element)
numbers = [1, 2, 3, 4, 5]
print_elements(numbers)
```

OUTPUT:



```
Using for loop:
1
2
3
4
5

=== Code Execution Successful ===
```

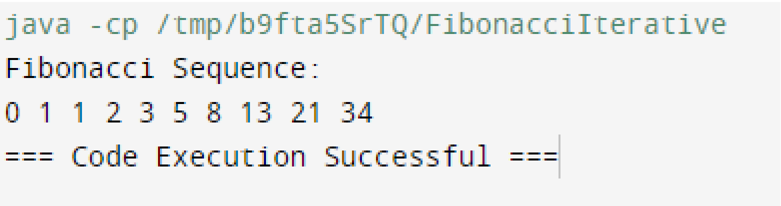
B] AIM:Implement a dynamic programming solution using arrays, such as the Fibonacci sequence or the Knapsack problem.

CODE:

java

```
public class FibonacciIterative {
    public static void main(String[] args) {
        int n = 10;
        printFibonacci(n);
    }
    public static void printFibonacci(int n) {
        if (n <= 0) {
            System.out.println("The number of terms should be positive.");
            return;
        }
        int a = 0, b = 1;
        System.out.println("Fibonacci Sequence:");
        System.out.print(a + " ");
        if (n > 1) {
            System.out.print(b + " ");
        }
        for (int i = 2; i < n; i++) {
            int next = a + b;
            System.out.print(next + " ");
            a = b;
            b = next;
        }
    }
}
```

OUTPUT:



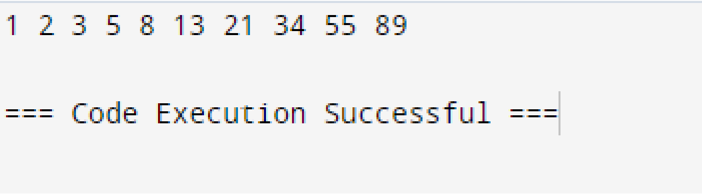
```
java -cp /tmp/b9fta5SrTQ/FibonacciIterative
Fibonacci Sequence:
0 1 1 2 3 5 8 13 21 34
=== Code Execution Successful ===
```

CODE:

python

```
n = 10
num1 = 0
num2 = 1
next_number = num2
count = 1

while count <= n:
        print(next_number, end=" ")
        count += 1
        num1, num2 = num2, next_number
        next_number = num1 + num2
print()
```

OUTPUT:

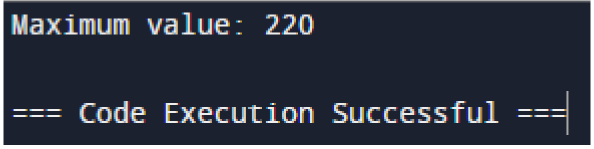


```
1 2 3 5 8 13 21 34 55 89

=== Code Execution Successful ===
```

OR

```
def knapSack(W, wt, val, n):
    dp = [[0] * (W + 1) for _ in range(n + 1)]

    for i in range(1, n + 1):
        for w in range(1, W + 1):
            if wt[i - 1] <= w:
                dp[i][w] = max(val[i - 1] + dp[i - 1][w - wt[i - 1]], dp[i - 1][w])
            else:
                dp[i][w] = dp[i - 1][w]

    return dp[n][W]

# Example data
val = [60, 100, 120]
wt = [10, 20, 30]
W = 50
```

```
n = len(val)

print("Maximum value:", knapSack(W, wt, val, n))
```



```
Maximum value: 220

=== Code Execution Successful ===
```

Practical No 5

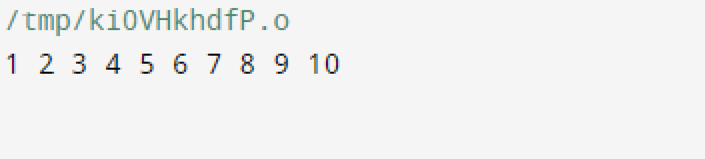
AIM: Write functions to dynamically allocate memory for arrays using pointers, and handle memory deallocation.

CODE:

C

```
#include <stdio.h>
#include <stdlib.h>
int* allocateAndInitializeArray(int size) {
    int* array = (int*)malloc(size * sizeof(int));
    if (array == NULL) {
        printf("Memory allocation failed\n");
        exit(1);
    }
    for (int i = 0; i < size; i++) {
        array[i] = i + 1; // Example initialization
    }
    return array;
}
void printArray(int* array, int size) {
    for (int i = 0; i < size; i++) {
        printf("%d ", array[i]);
    }
    printf("\n");
}
void deallocateArray(int* array) {
    free(array);
}
int main() {
    int size = 10;
    int* array = allocateAndInitializeArray(size);
    printArray(array, size);
    deallocateArray(array);
    return 0;
}
```

OUTPUT:



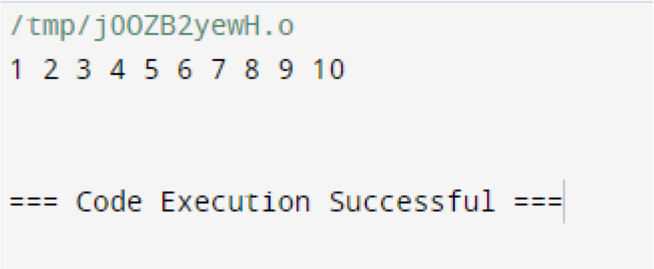
```
/tmp/kiOVHkhdfP.o
1 2 3 4 5 6 7 8 9 10
```

CODE:

C++

```
#include <iostream>
int* allocateAndInitializeArray(int size) {
    int* array = new int[size];
    for (int i = 0; i < size; i++) {
        array[i] = i + 1;
    }
    return array;
}
void printArray(int* array, int size) {
    for (int i = 0; i < size; i++) {
        std::cout << array[i] << " ";
    }
    std::cout << std::endl;
}
void deallocateArray(int* array) {
    delete[] array;
}
int main() {
    int size = 10;
    int* array = allocateAndInitializeArray(size);
    printArray(array, size);
    deallocateArray(array);
    return 0;
}
```

OUTPUT:



```
/tmp/jOOZB2yewH.o
1 2 3 4 5 6 7 8 9 10


=== Code Execution Successful ===
```

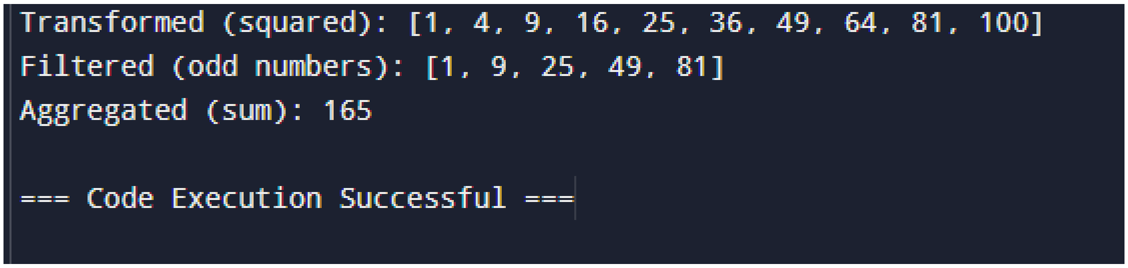
Practical No 6

AIM: a) Write functions using map, filter, and reduce to transform, filter, and aggregate data respectively.

CODE:

```
from functools import reduce
def transform_data(numbers):
    return list(map(lambda x: x ** 2, numbers))
def filter_data(numbers):
    return list(filter(lambda x: x % 2 != 0, numbers))
def aggregate_data(numbers):
    return reduce(lambda x, y: x + y, numbers)
if _name___== "_main_":
    numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
    transformed = transform_data(numbers)
    print(f"Transformed (squared): {transformed}")
    filtered = filter_data(transformed)
    print(f"Filtered (odd numbers): {filtered}")
    aggregated = aggregate_data(filtered)
    print(f"Aggregated (sum): {aggregated}")
```

OUTPUT:



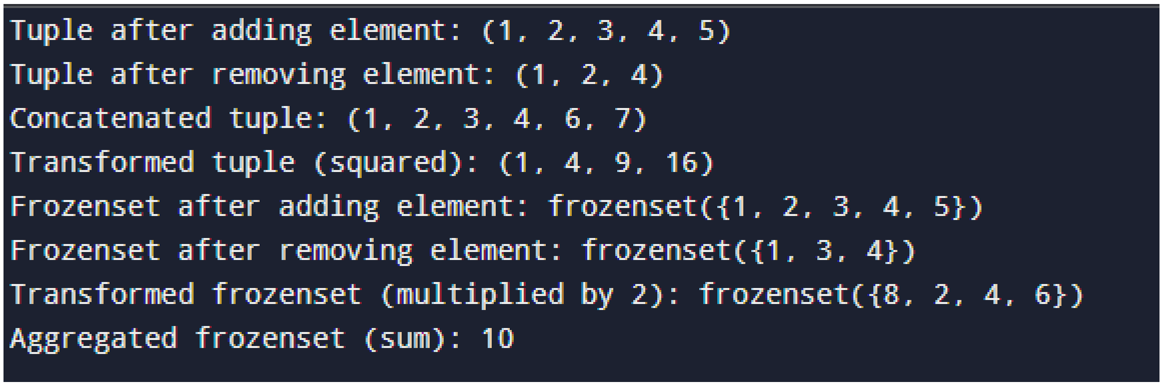
```
Transformed (squared): [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
Filtered (odd numbers): [1, 9, 25, 49, 81]
Aggregated (sum): 165

=== Code Execution Successful ===
```

b) Implement operations on immutable data structures (e.g., tuples, frozen sets) using functional programming principles.

CODE:

```
from functools import reduce
def add_to_tuple(tup, elem):
    return tup + (elem,)
def remove_from_tuple(tup, elem):
    return tuple(x for x in tup if x != elem)
def concatenate_tuples(tup1, tup2):
    return tup1 + tup2
def transform_tuple(tup):
    return tuple(map(lambda x: x**2, tup))
def add_to_frozenset(fset, elem):
    return fset.union([elem])
def remove_from_frozenset(fset, elem):
    return frozenset(x for x in fset if x != elem)
def transform_frozenset(fset):
    return frozenset(map(lambda x: x * 2, fset))
def aggregate_frozenset(fset):
    return reduce(lambda x, y: x + y, fset)
if _name___== "_main_":
    my_tuple = (1, 2, 3, 4)
    new_tuple = add_to_tuple(my_tuple, 5)
    print(f"Tuple after adding element: {new_tuple}")
    new_tuple = remove_from_tuple(my_tuple, 3)
    print(f"Tuple after removing element: {new_tuple}")
    another_tuple = (6, 7)
    concatenated_tuple = concatenate_tuples(my_tuple, another_tuple)
    print(f"Concatenated tuple: {concatenated_tuple}")
    transformed_tuple = transform_tuple(my_tuple)
    print(f"Transformed tuple (squared): {transformed_tuple}")
    my_frozenset = frozenset([1, 2, 3, 4])
    new_frozenset = add_to_frozenset(my_frozenset, 5)
    print(f"Frozenset after adding element: {new_frozenset}")
    new_frozenset = remove_from_frozenset(my_frozenset, 2)
    print(f"Frozenset after removing element: {new_frozenset}")
    transformed_frozenset = transform_frozenset(my_frozenset)
    print(f"Transformed frozenset (multiplied by 2): {transformed_frozenset}")
    aggregated_value = aggregate_frozenset(my_frozenset)
    print(f"Aggregated frozenset (sum): {aggregated_value}")
```

OUTPUT:



```
Tuple after adding element: (1, 2, 3, 4, 5)
Tuple after removing element: (1, 2, 4)
Concatenated tuple: (1, 2, 3, 4, 6, 7)
Transformed tuple (squared): (1, 4, 9, 16)
Frozenset after adding element: frozenset({1, 2, 3, 4, 5})
Frozenset after removing element: frozenset({1, 3, 4})
Transformed frozenset (multiplied by 2): frozenset({8, 2, 4, 6})
Aggregated frozenset (sum): 10
```

Practical No 7

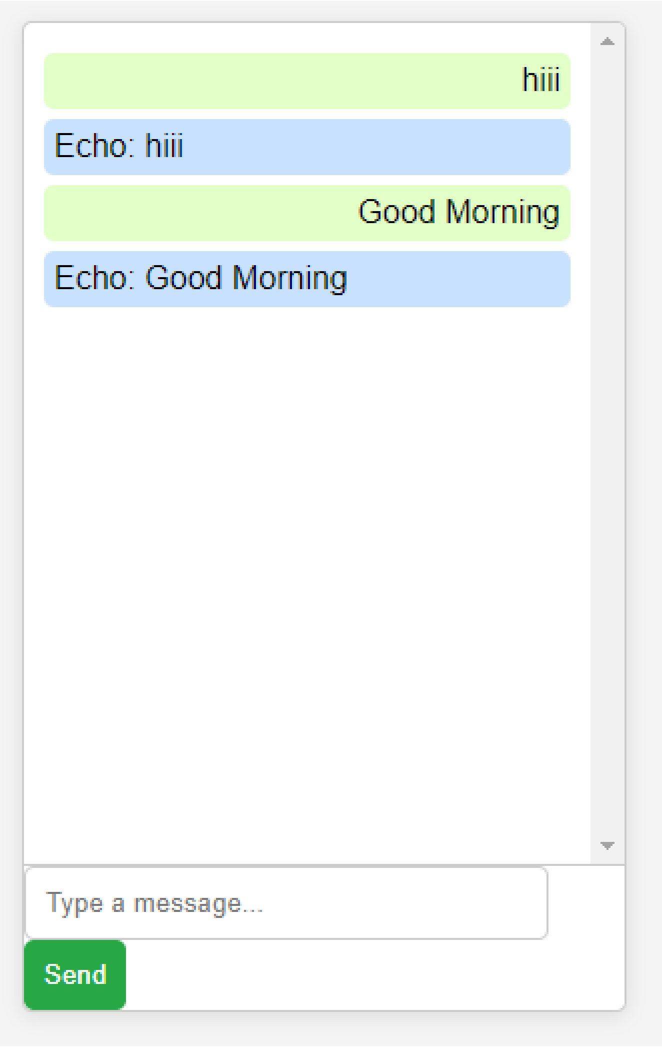
AIM: To develop a real-time chat application using event-driven programming principles, ensuring that messages are sent and received instantly between clients without refreshing the page.

CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Simple Chat App</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 0;
        }


        .chat-container {
            width: 300px;
            margin: 50px auto;
            border: 1px solid #ccc;
            border-radius: 5px;
            background: #fff;
            box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
            overflow: hidden;
        }


        .messages {
            height: 400px;
            padding: 10px;
            overflow-y: scroll;
            border-bottom: 1px solid #ccc;
        }


        .message {
            margin: 5px 0;
            padding: 5px;
            border-radius: 5px;
        }


        .message.sent {
            background-color: #e1ffc7;
            text-align: right;
        }


        .message.received {
            background-color: #c7e1ff;
            text-align: left;
        }


        input[type="text"] {
            width: calc(100% - 60px);
            padding: 10px;
            border: 1px solid #ccc;
            border-radius: 5px;
        }


        button {
            padding: 10px;
            border: none;
            background-color: #28a745;
            color: white;
            border-radius: 5px;
            cursor: pointer;
        }


        button:hover {
            background-color: #218838;
        }
    </style>
</head>
<body>
    <div class="chat-container">
        <div class="messages" id="messages"></div>
        <input type="text" id="messageInput" placeholder="Type a message..." />
        <button id="sendButton">Send</button>
    </div>
    <script>
```

```
        document.getElementById('sendButton').addEventListener('click', function() {
            const messageInput = document.getElementById('messageInput');
            const messageText = messageInput.value;


            if (messageText.trim() === '') return;


            // Create a new message element
            const messageElement = document.createElement('div');
            messageElement.classList.add('message', 'sent');
            messageElement.textContent = messageText;


            // Append the message to the messages container
            document.getElementById('messages').appendChild(messageElement);


            // Clear the input
            messageInput.value = '';


            // Scroll to the bottom of the messages
            document.getElementById('messages').scrollTop =
document.getElementById('messages').scrollHeight;


            // Simulate receiving a message after a short delay
            setTimeout(() => {
                const receivedMessageElement = document.createElement('div');
                receivedMessageElement.classList.add('message', 'received');
                receivedMessageElement.textContent = "Echo: " + messageText;


                document.getElementById('messages').appendChild(receivedMessageElement);
                document.getElementById('messages').scrollTop =
document.getElementById('messages').scrollHeight;
            }, 1000);
        });


        // Optional: Send message on pressing Enter
        document.getElementById('messageInput').addEventListener('keypress', function(e) {
            if (e.key === 'Enter') {
                document.getElementById('sendButton').click();
            }
        });
    </script>
</body>
</html>
```

OUTPUT:



```
hiii
Echo: hiii
Good Morning
Echo: Good Morning
```

Practical No 8

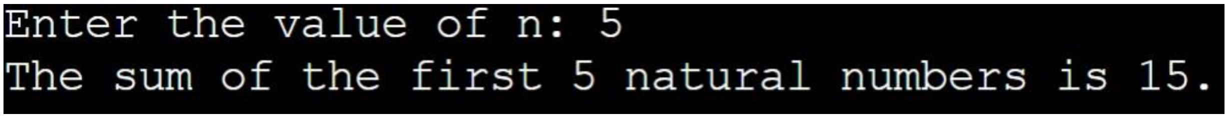
AIM: Implement using Go Programming :

a) Calculate sum of first n numbers.

CODE:

```
package main
import (
        "fmt"
)
// Function to calculate the sum of the first n natural numbers
func sumOfFirstNNumbers(n int) int {
        sum := 0
        for i := 1; i <= n; i++ {
                sum += i
        }
        return sum
}
func main() {
        var n int
        fmt.Print("Enter the value of n: ")
        fmt.Scan(&n)
        if n < 1 {
                fmt.Println("Please enter a positive integer greater than 0.")
                return
        }
        result := sumOfFirstNNumbers(n)
        fmt.Printf("The sum of the first %d natural numbers is %d.\n", n, result)
}
```

OUTPUT:



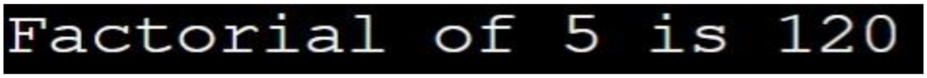
```
Enter the value of n: 5
The sum of the first 5 natural numbers is 15.
```

b)Recursive function to find factorial of a number.

CODE:

```
package main

import (
    "fmt"
)

// Recursive function to find factorial
func factorial(n int) int {
    if n == 0 {
        return 1
    }
    return n * factorial(n-1)
}

func main() {
    number := 5
    fmt.Printf("Factorial of %d is %d\n", number, factorial(number))
}
```

OUTPUT:



```
Factorial of 5 is 120
```

Practical No 9

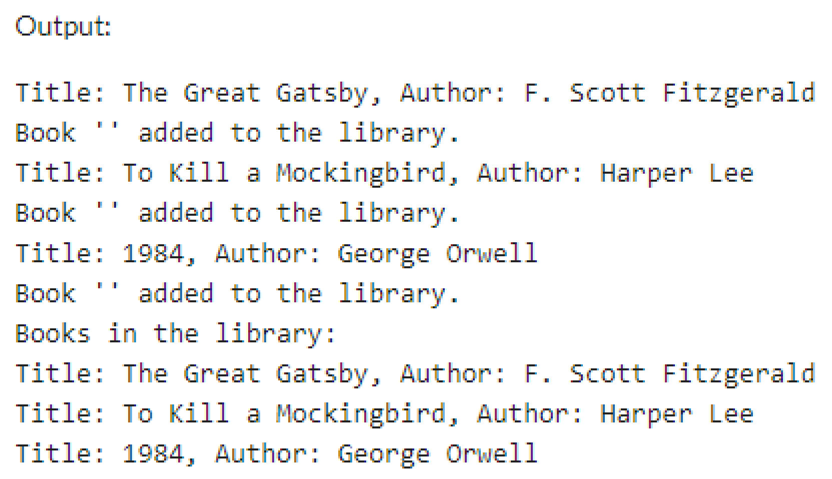
AIM: Implement using Ruby programming :

a) Define classes and create objects with attributes and methods.

CODE:

```
class Book
  def initialize(title, author)
    @title = title
    @author = author
  end
  def display_details
    puts "Title: #{@title}, Author: #{@author}"
  end
end
class Library

  def initialize
  @books = []
  end
  def add_book(book)
    @books << book
    puts "Book '#{book.display_details}' added to the library."
  end
  def list_books
    if @books.empty?
      puts "No books in the library."
    else
      puts "Books in the library:"
      @books.each do |book|
      book.display_details
      end
end
end
end
book1 = Book.new("The Great Gatsby", "F. Scott Fitzgerald")
book2 = Book.new("To Kill a Mockingbird", "Harper Lee")
book3 = Book.new("1984", "George Orwell")
library = Library.new
library.add_book(book1)
library.add_book(book2)
library.add_book(book3)
library.list_books
```

OUTPUT:



```
Output:

Title: The Great Gatsby, Author: F. Scott Fitzgerald
Book '' added to the library.
Title: To Kill a Mockingbird, Author: Harper Lee
Book '' added to the library.
Title: 1984, Author: George Orwell
Book '' added to the library.
Books in the library:
Title: The Great Gatsby, Author: F. Scott Fitzgerald
Title: To Kill a Mockingbird, Author: Harper Lee
Title: 1984, Author: George Orwell
```

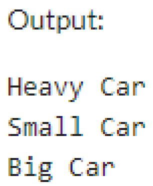
b) Implement inheritance and demonstrate polymorphic behavior.

CODE:

```
class Vehicle
        def tyreType
                puts "Heavy Car"
        end
end

class Car < Vehicle
        def tyreType
                puts "Small Car"
        end
end

class Truck < Vehicle
        def tyreType
                puts "Big Car"
        end
end

vehicle = Vehicle.new
vehicle.tyreType()

vehicle = Car.new
vehicle.tyreType()
```

```
vehicle = Truck.new
vehicle.tyreType()
```

OUTPUT:



Output:

```
Heavy Car
Small Car
Big Car
```

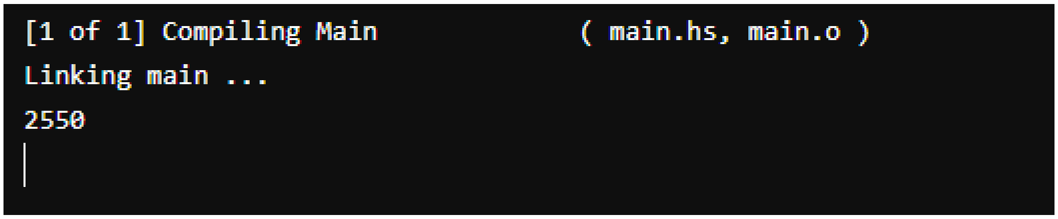
Practical No 10

AIM: Implement a functional program to calculate the sum of squares of even numbers from a list using Haskell.

CODE:

```
main :: IO()
main = print (sum [x | x <- [1..100], x `mod` 2 == 0])
```

OUTPUT:



```
[1 of 1] Compiling Main             ( main.hs, main.o )
Linking main ...
2550
```