Assignment 1

Task 1.1:

Source Code:

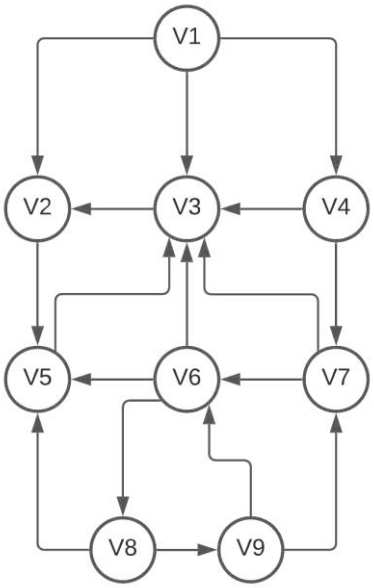
```
import java.util.Scanner;
import java.util.Stack;
class CheckInitiatorNode
{
    int no_vertex, adjacency_matrix[][], visited[];
    Stack<Integer> stack = new Stack<Integer>();

    public CheckInitiatorNode(int n)
    {
        no_vertex=n;
        adjacency_matrix=new int[no_vertex][no_vertex];
        visited=new int[no_vertex];
    }

    public void InputAdjacencyMatrix()
    {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the Adjacency Matrix:");
        int i,j,input;
        for(i=0;i<no_vertex;i++)
        {
            for(j=0;j<no_vertex;j++)
            {
                // Input should either be 0 indicating no edge or 1 indicating
existence of an edge
                System.out.print("Enter the directed edge (0/1) between
vertices V"+(i+1)+" and V"+(j+1)+" : ");
                input=scanner.nextInt();
                if(input!=1 && input!=0)
                {
                    System.out.println("Invalid Input!");
                    System.exit(0);
                }
                adjacency_matrix[i][j]=input;
            }
        }
    }

    public void DisplayAdjacencyMatrix()
    {
        System.out.println("The given Adjacency Matrix:");
        int i,j;
        for(i=0;i<no_vertex;i++)
        {
            for(j=0;j<no_vertex;j++)
```

```
            {
                System.out.print(adjacency_matrix[i][j]+"  ");
            }
            System.out.println();
        }
    }

    public void CheckInitiator()
    {
        Scanner scanner = new Scanner(System.in);
        int node,i,j;
        // Input should be V1/V2/... in this format
        System.out.print("\nEnter the node to be checked (V1/V2/..): ");
        String n=scanner.nextLine();
        node=Character.getNumericValue(n.charAt(1));
        if(node>no_vertex)
        {
            System.out.println("Invalid Input!");
            System.exit(0);
        }

        stack.push(node-1);
        visited[node-1]=1;
        Integer removed = (Integer) stack.pop();
        i=removed;
        while(true)
        {
            for(j=0;j<no_vertex;j++)
            {
                if(adjacency_matrix[i][j]==1 && visited[j]==0)
                {
                    stack.push(j);
                    visited[j]=1;
                }
            }
            //If the stack is empty then we have to check whether the all the
elements in the visited array are 1
            if(stack.isEmpty()==true)
                break;
            removed=(Integer) stack.pop();
            i=removed;
        }
        for(i=0;i<no_vertex;i++)
        {
            // If all the elements in the visited array are 1 then that
corresponding node is an initiator node, otherwise not
            if(visited[i]==0)
            {
```

```
                System.out.println("The given node V"+node+" cannot be an
initiator node");
                return;
            }
        }
        System.out.println("The given node V"+node+" can be an initiator
node");
    }

    public static void main(String args[])
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("\nEnter the number of vertices: ");
        int n=scanner.nextInt();
        if(n<1)
        {
            System.out.println("Invalid Input!");
            return;
        }
        CheckInitiatorNode obj = new CheckInitiatorNode(n);

        obj.InputAdjacencyMatrix();
        obj.DisplayAdjacencyMatrix();
        obj.CheckInitiator();

    }
}
```

Set 1:

Dataset Used

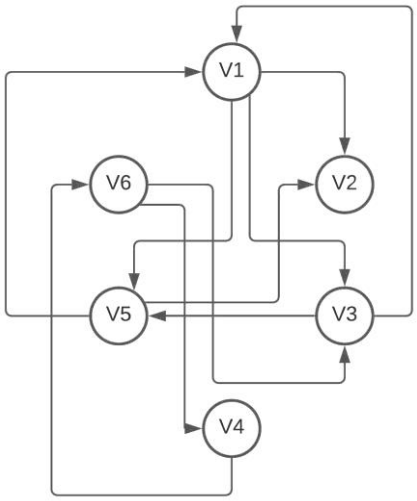


Output Obtained:

```
PS C:\Users\debal\Documents\Assignments\msc-sem3-AOS> java CheckInitiatorN

Enter the number of vertices: 9
Enter the Adjacency Matrix:
Enter the directed edge (0/1) between vertices V1 and V1 : 0
Enter the directed edge (0/1) between vertices V1 and V2 : 1
Enter the directed edge (0/1) between vertices V1 and V3 : 1
Enter the directed edge (0/1) between vertices V1 and V4 : 1
Enter the directed edge (0/1) between vertices V1 and V5 : 0
Enter the directed edge (0/1) between vertices V1 and V6 : 0
Enter the directed edge (0/1) between vertices V1 and V7 : 0
Enter the directed edge (0/1) between vertices V1 and V8 : 0
Enter the directed edge (0/1) between vertices V1 and V9 : 0
Enter the directed edge (0/1) between vertices V2 and V1 : 0
Enter the directed edge (0/1) between vertices V2 and V2 : 0
Enter the directed edge (0/1) between vertices V2 and V3 : 0
Enter the directed edge (0/1) between vertices V2 and V4 : 0
Enter the directed edge (0/1) between vertices V2 and V5 : 1
Enter the directed edge (0/1) between vertices V2 and V6 : 0
Enter the directed edge (0/1) between vertices V2 and V7 : 0
Enter the directed edge (0/1) between vertices V2 and V8 : 0
Enter the directed edge (0/1) between vertices V2 and V9 : 0
Enter the directed edge (0/1) between vertices V3 and V1 : 0
Enter the directed edge (0/1) between vertices V3 and V2 : 1
Enter the directed edge (0/1) between vertices V3 and V3 : 0
Enter the directed edge (0/1) between vertices V3 and V4 : 0
Enter the directed edge (0/1) between vertices V3 and V5 : 0
Enter the directed edge (0/1) between vertices V3 and V6 : 0
Enter the directed edge (0/1) between vertices V3 and V7 : 0
Enter the directed edge (0/1) between vertices V3 and V8 : 0
Enter the directed edge (0/1) between vertices V3 and V9 : 0
Enter the directed edge (0/1) between vertices V4 and V1 : 0
Enter the directed edge (0/1) between vertices V4 and V2 : 0
Enter the directed edge (0/1) between vertices V4 and V3 : 1
Enter the directed edge (0/1) between vertices V4 and V4 : 0
Enter the directed edge (0/1) between vertices V4 and V5 : 0
Enter the directed edge (0/1) between vertices V4 and V6 : 0
Enter the directed edge (0/1) between vertices V4 and V7 : 1
Enter the directed edge (0/1) between vertices V4 and V8 : 0
Enter the directed edge (0/1) between vertices V4 and V9 : 0
Enter the directed edge (0/1) between vertices V5 and V1 : 0
Enter the directed edge (0/1) between vertices V5 and V2 : 0
Enter the directed edge (0/1) between vertices V5 and V3 : 1
Enter the directed edge (0/1) between vertices V5 and V4 : 0
Enter the directed edge (0/1) between vertices V5 and V5 : 0
Enter the directed edge (0/1) between vertices V5 and V6 : 0
Enter the directed edge (0/1) between vertices V5 and V7 : 0
Enter the directed edge (0/1) between vertices V5 and V8 : 0
Enter the directed edge (0/1) between vertices V5 and V9 : 0
Enter the directed edge (0/1) between vertices V6 and V1 : 0
Enter the directed edge (0/1) between vertices V6 and V2 : 0
Enter the directed edge (0/1) between vertices V6 and V3 : 1
Enter the directed edge (0/1) between vertices V6 and V4 : 0
Enter the directed edge (0/1) between vertices V6 and V5 : 1
Enter the directed edge (0/1) between vertices V6 and V6 : 0
Enter the directed edge (0/1) between vertices V6 and V7 : 0
Enter the directed edge (0/1) between vertices V6 and V8 : 1
Enter the directed edge (0/1) between vertices V6 and V9 : 0
Enter the directed edge (0/1) between vertices V7 and V1 : 0
Enter the directed edge (0/1) between vertices V7 and V2 : 0
Enter the directed edge (0/1) between vertices V8 and V9 : 1
Enter the directed edge (0/1) between vertices V9 and V1 : 0
Enter the directed edge (0/1) between vertices V9 and V2 : 0
Enter the directed edge (0/1) between vertices V9 and V3 : 0
Enter the directed edge (0/1) between vertices V9 and V4 : 0
Enter the directed edge (0/1) between vertices V9 and V5 : 0
Enter the directed edge (0/1) between vertices V9 and V6 : 1
Enter the directed edge (0/1) between vertices V9 and V7 : 1
Enter the directed edge (0/1) between vertices V9 and V8 : 0
Enter the directed edge (0/1) between vertices V9 and V9 : 0
The given Adjacency Matrix:
0  1  1  1  0  0  0  0  0
0  0  0  0  1  0  0  0  0
0  1  0  0  0  0  0  0  0
0  0  1  0  0  0  1  0  0
0  0  1  0  0  0  0  0  0
0  0  1  0  1  0  0  1  0
0  0  1  0  0  1  0  0  0
0  0  0  0  1  0  0  0  1
0  0  0  0  0  1  1  0  0

Enter the node to be checked (V1/V2/..): V1
The given node V1 can be an initiator node
```

Set 2:

Dataset Used



Output Obtained:

```
PS C:\Users\debal\Documents\Assignments\msc-sem3-AOS> java CheckInitiatorNode

Enter the number of vertices: 6
Enter the Adjacency Matrix:
Enter the directed edge (0/1) between vertices V1 and V1 : 0
Enter the directed edge (0/1) between vertices V1 and V2 : 1
Enter the directed edge (0/1) between vertices V1 and V3 : 1
Enter the directed edge (0/1) between vertices V1 and V4 : 0
Enter the directed edge (0/1) between vertices V1 and V5 : 1
Enter the directed edge (0/1) between vertices V1 and V6 : 0
Enter the directed edge (0/1) between vertices V2 and V1 : 0
Enter the directed edge (0/1) between vertices V2 and V2 : 0
Enter the directed edge (0/1) between vertices V2 and V3 : 0
Enter the directed edge (0/1) between vertices V2 and V4 : 0
Enter the directed edge (0/1) between vertices V2 and V5 : 0
Enter the directed edge (0/1) between vertices V2 and V6 : 0
Enter the directed edge (0/1) between vertices V3 and V1 : 1
Enter the directed edge (0/1) between vertices V3 and V2 : 0
Enter the directed edge (0/1) between vertices V3 and V3 : 0
Enter the directed edge (0/1) between vertices V3 and V4 : 0
Enter the directed edge (0/1) between vertices V3 and V5 : 1
Enter the directed edge (0/1) between vertices V3 and V6 : 0
Enter the directed edge (0/1) between vertices V4 and V1 : 0
Enter the directed edge (0/1) between vertices V4 and V2 : 0
Enter the directed edge (0/1) between vertices V4 and V3 : 0
Enter the directed edge (0/1) between vertices V4 and V4 : 0
Enter the directed edge (0/1) between vertices V4 and V5 : 0
Enter the directed edge (0/1) between vertices V4 and V6 : 1
Enter the directed edge (0/1) between vertices V5 and V1 : 1
Enter the directed edge (0/1) between vertices V5 and V2 : 1
Enter the directed edge (0/1) between vertices V5 and V3 : 0
Enter the directed edge (0/1) between vertices V5 and V4 : 0
Enter the directed edge (0/1) between vertices V5 and V5 : 0
Enter the directed edge (0/1) between vertices V5 and V6 : 0
Enter the directed edge (0/1) between vertices V6 and V1 : 0
Enter the directed edge (0/1) between vertices V6 and V2 : 0
Enter the directed edge (0/1) between vertices V6 and V3 : 1
Enter the directed edge (0/1) between vertices V6 and V4 : 1
Enter the directed edge (0/1) between vertices V6 and V5 : 0
Enter the directed edge (0/1) between vertices V6 and V6 : 0
The given Adjacency Matrix:
0  1  1  0  1  0
0  0  0  0  0  0
1  0  0  0  1  0
0  0  0  0  0  1
1  1  0  0  0  0
0  0  1  1  0  0

Enter the node to be checked (V1/V2/..): V4
The given node V4 can be an initiator node
```

Task 1.2:

Source Code:

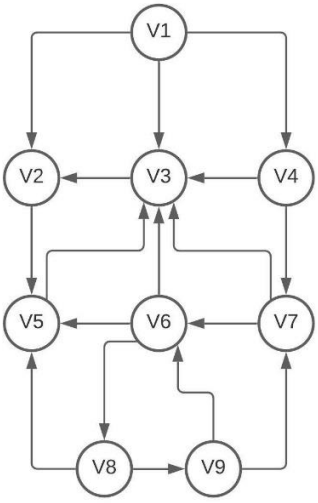
```
import java.util.Scanner;
import java.util.Stack;
class FindInitiatorNode
{
    int no_vertex, adjacency_matrix[][], visited[];
    Stack<Integer> stack = new Stack<Integer>();

    public FindInitiatorNode(int n)
    {
        no_vertex=n;
        adjacency_matrix=new int[no_vertex][no_vertex];
        visited=new int[no_vertex];
    }

    public void InputAdjacencyMatrix()
    {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the Adjacency Matrix:");
        int i,j,input;
        for(i=0;i<no_vertex;i++)
        {
            for(j=0;j<no_vertex;j++)
            {
                // Input should either be 0 indicating no edge or 1 indicating
existence of an edge
                System.out.print("Enter the directed edge (0/1) between
vertices V"+(i+1)+" and V"+(j+1)+" : ");
                input=scanner.nextInt();
                if(input!=1 && input!=0)
                {
                    System.out.println("Invalid Input!");
                    System.exit(0);
                }
                adjacency_matrix[i][j]=input;
            }
        }
    }

    public void DisplayAdjacencyMatrix()
    {
        System.out.println("The given Adjacency Matrix:");
        int i,j;
        for(i=0;i<no_vertex;i++)
        {
            for(j=0;j<no_vertex;j++)
            {
```

```
                System.out.print(adjacency_matrix[i][j]+" ");
            }
            System.out.println();
        }
    }

    public void FindInitiator()
    {
        int node,i,j,k;

        for(k=0;k<no_vertex;k++)
        {
            for(i=0;i<no_vertex;i++)
            {
                visited[i]=0;
            }
            i=k;
            stack.push(i);
            visited[i]=1;

            Integer removed = (Integer) stack.pop();
            while(true)
            {
                for(j=0;j<no_vertex;j++)
                {
                    if(adjacency_matrix[i][j]==1 && visited[j]==0)
                    {
                        stack.push(j);
                        visited[j]=1;
                    }
                }
                if(this.checkVisitedArray()==true)
                {
                    System.out.print("V"+(k+1)+" ");
                    break;
                }
                if(stack.isEmpty()==true)
                {
                   break;
                }
                removed=(Integer) stack.pop();
                i=removed;
            }
        }
    }

    public boolean checkVisitedArray()
    {
```

```
        for(int i=0;i<no_vertex;i++)
        {
            if(visited[i]==0)
            {
                return false;
            }
        }
        return true;
    }

    public static void main(String args[])
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("\nEnter the number of vertices: ");
        int n=scanner.nextInt();
        if(n<1)
        {
            System.out.println("Invalid Input!");
            return;
        }
        FindInitiatorNode obj = new  FindInitiatorNode(n);

        obj.InputAdjacencyMatrix();
        obj.DisplayAdjacencyMatrix();

        System.out.print("\nThe possible initiator nodes are : ");
        obj.FindInitiator();

    }
}
```

Set 1:

Dataset Used



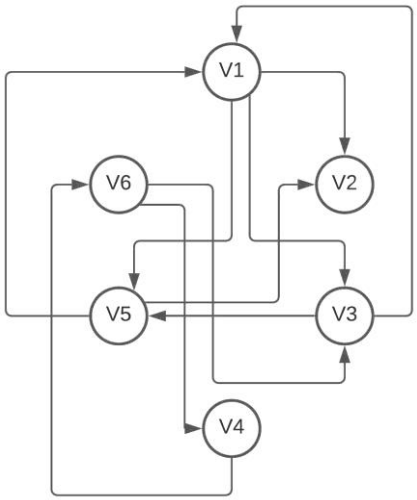
Output Obtained:

```
PS C:\Users\debal\Documents\Assignments\msc-sem3-AOS> java FindInitiatorNod

Enter the number of vertices: 9
Enter the Adjacency Matrix:
Enter the directed edge (0/1) between vertices V1 and V1 : 0
Enter the directed edge (0/1) between vertices V1 and V2 : 1
Enter the directed edge (0/1) between vertices V1 and V3 : 1
Enter the directed edge (0/1) between vertices V1 and V4 : 1
Enter the directed edge (0/1) between vertices V1 and V5 : 0
Enter the directed edge (0/1) between vertices V1 and V6 : 0
Enter the directed edge (0/1) between vertices V1 and V7 : 0
Enter the directed edge (0/1) between vertices V1 and V8 : 0
Enter the directed edge (0/1) between vertices V1 and V9 : 0
Enter the directed edge (0/1) between vertices V2 and V1 : 0
Enter the directed edge (0/1) between vertices V2 and V2 : 0
Enter the directed edge (0/1) between vertices V2 and V3 : 0
Enter the directed edge (0/1) between vertices V2 and V4 : 0
Enter the directed edge (0/1) between vertices V2 and V5 : 1
Enter the directed edge (0/1) between vertices V2 and V6 : 0
Enter the directed edge (0/1) between vertices V2 and V7 : 0
Enter the directed edge (0/1) between vertices V2 and V8 : 0
Enter the directed edge (0/1) between vertices V2 and V9 : 0
Enter the directed edge (0/1) between vertices V3 and V1 : 0
Enter the directed edge (0/1) between vertices V3 and V2 : 1
Enter the directed edge (0/1) between vertices V3 and V3 : 0
Enter the directed edge (0/1) between vertices V3 and V4 : 0
Enter the directed edge (0/1) between vertices V3 and V5 : 0
Enter the directed edge (0/1) between vertices V3 and V6 : 0
Enter the directed edge (0/1) between vertices V3 and V7 : 0
Enter the directed edge (0/1) between vertices V3 and V8 : 0
Enter the directed edge (0/1) between vertices V3 and V9 : 0
Enter the directed edge (0/1) between vertices V4 and V1 : 0
Enter the directed edge (0/1) between vertices V4 and V2 : 0
Enter the directed edge (0/1) between vertices V4 and V3 : 1
Enter the directed edge (0/1) between vertices V4 and V4 : 0
Enter the directed edge (0/1) between vertices V4 and V5 : 0
Enter the directed edge (0/1) between vertices V4 and V6 : 0
Enter the directed edge (0/1) between vertices V4 and V7 : 1
Enter the directed edge (0/1) between vertices V4 and V8 : 0
Enter the directed edge (0/1) between vertices V4 and V9 : 0
Enter the directed edge (0/1) between vertices V5 and V1 : 0
Enter the directed edge (0/1) between vertices V5 and V2 : 0
Enter the directed edge (0/1) between vertices V5 and V3 : 1
Enter the directed edge (0/1) between vertices V5 and V4 : 0
Enter the directed edge (0/1) between vertices V5 and V5 : 0
Enter the directed edge (0/1) between vertices V5 and V6 : 0
Enter the directed edge (0/1) between vertices V5 and V7 : 0
Enter the directed edge (0/1) between vertices V5 and V8 : 0
Enter the directed edge (0/1) between vertices V5 and V9 : 0
Enter the directed edge (0/1) between vertices V6 and V1 : 0
Enter the directed edge (0/1) between vertices V6 and V2 : 0
Enter the directed edge (0/1) between vertices V6 and V3 : 1
Enter the directed edge (0/1) between vertices V6 and V4 : 0
Enter the directed edge (0/1) between vertices V6 and V5 : 1
Enter the directed edge (0/1) between vertices V6 and V6 : 0
Enter the directed edge (0/1) between vertices V6 and V7 : 0
Enter the directed edge (0/1) between vertices V6 and V8 : 1
Enter the directed edge (0/1) between vertices V6 and V9 : 0
Enter the directed edge (0/1) between vertices V7 and V1 : 0
Enter the directed edge (0/1) between vertices V7 and V2 : 0
Enter the directed edge (0/1) between vertices V7 and V3 : 1
Enter the directed edge (0/1) between vertices V7 and V4 : 0
Enter the directed edge (0/1) between vertices V7 and V5 : 0
Enter the directed edge (0/1) between vertices V7 and V6 : 1
Enter the directed edge (0/1) between vertices V7 and V7 : 0
Enter the directed edge (0/1) between vertices V7 and V8 : 0
Enter the directed edge (0/1) between vertices V7 and V9 : 0
Enter the directed edge (0/1) between vertices V8 and V1 : 0
Enter the directed edge (0/1) between vertices V8 and V2 : 0
Enter the directed edge (0/1) between vertices V8 and V3 : 0
Enter the directed edge (0/1) between vertices V8 and V4 : 0
Enter the directed edge (0/1) between vertices V8 and V5 : 1
Enter the directed edge (0/1) between vertices V8 and V6 : 0
Enter the directed edge (0/1) between vertices V8 and V7 : 0
Enter the directed edge (0/1) between vertices V8 and V8 : 0
Enter the directed edge (0/1) between vertices V8 and V9 : 1
Enter the directed edge (0/1) between vertices V9 and V1 : 0
Enter the directed edge (0/1) between vertices V9 and V2 : 0
Enter the directed edge (0/1) between vertices V9 and V3 : 0
Enter the directed edge (0/1) between vertices V9 and V4 : 0
Enter the directed edge (0/1) between vertices V9 and V5 : 0
```

```
Enter the directed edge (0/1) between vertices V9 and V6 : 1
Enter the directed edge (0/1) between vertices V9 and V7 : 1
Enter the directed edge (0/1) between vertices V9 and V8 : 0
Enter the directed edge (0/1) between vertices V9 and V9 : 0
The given Adjacency Matrix:
0 1 1 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0
0 1 0 0 0 0 0 0 0
0 0 1 0 0 0 1 0 0
0 0 1 0 0 0 0 0 0
0 0 1 0 1 0 0 1 0
0 0 1 0 0 1 0 0 0
0 0 0 0 1 0 0 0 1
0 0 0 0 0 1 1 0 0

The possible initiator nodes are : V1
```

Set 2:

Dataset Used



Output Obtained:

```
PS C:\Users\debal\Documents\Assignments\msc-sem3-AOS> java FindInitiatorNode

Enter the number of vertices: 6
Enter the Adjacency Matrix:
Enter the directed edge (0/1) between vertices V1 and V1 : 0
Enter the directed edge (0/1) between vertices V1 and V2 : 1
Enter the directed edge (0/1) between vertices V1 and V3 : 1
Enter the directed edge (0/1) between vertices V1 and V4 : 0
Enter the directed edge (0/1) between vertices V1 and V5 : 1
Enter the directed edge (0/1) between vertices V1 and V6 : 0
Enter the directed edge (0/1) between vertices V2 and V1 : 0
Enter the directed edge (0/1) between vertices V3 and V1 : 1
Enter the directed edge (0/1) between vertices V3 and V2 : 0
Enter the directed edge (0/1) between vertices V3 and V3 : 0
Enter the directed edge (0/1) between vertices V3 and V4 : 0
Enter the directed edge (0/1) between vertices V3 and V5 : 1
Enter the directed edge (0/1) between vertices V3 and V6 : 0
Enter the directed edge (0/1) between vertices V4 and V1 : 0
Enter the directed edge (0/1) between vertices V4 and V2 : 0
Enter the directed edge (0/1) between vertices V4 and V3 : 0
Enter the directed edge (0/1) between vertices V4 and V4 : 0
Enter the directed edge (0/1) between vertices V4 and V5 : 0
Enter the directed edge (0/1) between vertices V4 and V6 : 1
Enter the directed edge (0/1) between vertices V5 and V1 : 1
Enter the directed edge (0/1) between vertices V5 and V2 : 1
Enter the directed edge (0/1) between vertices V5 and V3 : 0
Enter the directed edge (0/1) between vertices V5 and V4 : 0
Enter the directed edge (0/1) between vertices V5 and V5 : 0
Enter the directed edge (0/1) between vertices V5 and V6 : 0
Enter the directed edge (0/1) between vertices V6 and V1 : 0
Enter the directed edge (0/1) between vertices V6 and V2 : 0
Enter the directed edge (0/1) between vertices V6 and V3 : 1
Enter the directed edge (0/1) between vertices V6 and V4 : 1
Enter the directed edge (0/1) between vertices V6 and V5 : 0
Enter the directed edge (0/1) between vertices V6 and V6 : 0
The given Adjacency Matrix:
0 1 1 0 1 0
0 0 0 0 0 0
1 0 0 0 1 0
0 0 0 0 0 1
1 1 0 0 0 0
0 0 1 1 0 0

The possible initiator nodes are : V4 V6
```