**Name:Ajinkya Madhav Hadole**

**Roll No: 13**

**Subject: Data Structure and Algorithm**

**Class: MCA**

**Sem-1**

# DSA PRACTICAL JOURNAL

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.**  **No** | **Title** | **Remark** | **Signature** |
|  | Write a program to create singly linear |  |  |
| **1** | linked list for insert delete search print  operation using menu driven program. |  |  |
|  | Write a program to create doubly circular |  |  |
| **2** | linked list for insert delete search print operation using menu driven program. |  |  |
| **3** | Write a program to check the parenthesis of expression is well formed ness or not**.** |  |  |
|  | Write a program to create stack using linked |  |  |
| **4** | list and manipulate it using menu driven cod  e |  |  |
| **5** | Write a program to create CQ using array  and manipulate it using menu driven code. |  |  |
| **6** | Write a program to reverse the given queue. |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Write a program to create Binary search tree |  |  |
| **7** | and traverse it using recursive preorder, inorder, postorder methods. |  |  |
|  | Write a program to implement graph using |  |  |
| **8** | adjacency matrix and traverse it using BFT and  DFT. |  |  |
|  | Write a program to implement linear |  |  |
| **9** | probing hashing for insert and search operati on. |  |  |
| **10** | Write a program which implement the rain terraces to measure trapping rain water. |  |  |
| **11** | Writ progra e a m to implements to  implement Kruskal’s Algorithm  MST. |  |  |
| **12** | Writ progra  e a m to implements to    implement Prim’s Algorithm MST. |  |  |
|  | Write a program to implement binary search |  |  |
| **13** | algorithm using divide and conquer technique. |  |  |
| **14** | Write a program to implement TOWER OF  HANNOI problem. |  |  |
|  | Write a program to find longest common sub |  |  |
| **15** | sequence. |  |  |
|  | Write a program to implement regular |  |  |
| **16** | expression to check whether given pattern match with input string or not? |  |  |
| **17** | Write a program to implement N queen  Problem using backtracking approach. |  |  |

**1. Write a program to create singly linear linked list for insert delete search print operation using menu driven program.**

**Solution:**

**/\*HTML Source Code\*/**

<html>

<head>

<script type="text/javascript" src="SLL1.js" ></script>

</head>

<body>

Click here for the result

<input type="button" onclick="Hi()" value="Singly Linear Linked List"

/>

</body>

</html>

**/\*JavaScript Source Code\*/**

class Node

{

constructor(data,next=null)

{

this.data=data; this.next=next;

}

}

class LinkedList

{

constructor()

{

this.head=null; this.size=0;

}

create(n)

{

var newnode=null,oldnode=this.head;

var i,d;

for(i=0;i<n;i++)

{

d=parseInt(prompt("Enter the value for node"));

newnode=new Node(d);

if(this.head==null)

{

this.head=newnode;

oldnode=newnode;

}

else

{

oldnode.next=newnode;

oldnode=newnode;

}

}

this.size++;

}

insertFirst(data)

{

alert(" in insert.first..."); this.head=new Node(data,this.head); this.size++;

}

printAllNodes()

{

alert("in print"); let ptr=this.head;

alert(" in print Size=..."+this.size); document.write("Singly linear linked list :->>"); while(ptr)

{

alert("in print"+ptr.data);

console.log(ptr.data);

document.write(ptr.data+":->>");

ptr=ptr.next;

}

}

insertLast(data)

{

alert(" in insert.Last...");

let node=new Node(data);

let ptr; if(this.head==null)

{

this.head=node;

}

else

{

ptr=this.head; while(ptr.next) { ptr=ptr.next;

}

ptr.next=node;

}

this.size++;

}

insert(item,pos)

{

var i=1;

let newnode=null,ptr=null; newnode= new Node(item); if(newnode==null)

{

document.write("Linked list Overflow");

}

else

{

if(pos==1)

{

newnode.next=this.head; this.head=newnode;

}

else

{

ptr=this.head; i=1;

}

while((i<pos-1)&&(ptr!=null))

{ i++; ptr=ptr.next;

}

if(ptr!=null)

{

newnode.next=ptr.next; ptr.next=newnode;

}

}

this.size++;

}

del(pos)

{

var i=1,item=0; alert("within del...");

let delnode=null,ptr=null; if(this.head==null)

{

document.write("linked list underflow");

} else

{

if(pos==1)

{

alert("within del pos=1..."); delnode=this.head; this.head=this.head.next; item=delnode.data; document.write("deleted item="+item);

}

else

{

ptr=this.head;i=1;

Page | 6

while((i<pos-1)&&(ptr!=null))

{

i++; ptr=ptr.next;

}

if(ptr==null)

{

document.write("Invalid position");

}

else

{

delnode=ptr.next; item=delnode.data; ptr.next=delnode.next; document.write("deleted item="+item); }

}

} this.size--;

}

}

function Hi()

{

const l1=new LinkedList(); do alert("HELLO"); do{

ch=parseInt(prompt("1: create 2:insert 3:delete 4:print 5:exit")); switch(ch)

{ case 1:

var n=parseInt(prompt("ENTER TOTAL NODES")); l1.create(n); break; case 2:item=parseInt(prompt("ENTER INTEM TO INSERT"));

pos=parseInt(prompt("ENTER POSITION TO INSERT"));

l1.insert(item,pos); break; case 3:

Page | 7

pos=parseInt(prompt("ENTER POSITION TO DELETE")); l1.del(pos); break; case 4:

l1.printAllNodes();break; case 5:break;

default : alert("ENTER RIGHT CHOISE");

}

}while(ch!=5);

}

Output:

Singly linear linked list :->>10:->>20:->>30:->>40:->>50:->>Singly linear linked list :->>1000:->>10:->>20:->>30:->>40:->>50:->>deleted item=1000Singly linear linked list :->>10:->>20:->>30:->>40:->>50:->>

Page | 8

**2. Write a program to create doubly circular linked list for insert delete search print operation using menu driven program.**

**Solution:**

**/\*HTML Source Code\*/**

<html>

<head>

<script type="text/javascript" src="DCL.js" ></script>

</head>

<body>

Click here for the result

<input type="button" onclick="dcl()" value="Doubly Circular Linked

List" />

</body>

</html>

**/\*JavaScript Source Code\*/**class DENode

{

constructor(ec=0,nm=" ",s=0.0,prev=null,next=null)

{

this.prev=prev;

this.name=nm; this.code=ec; this.sal=s; this.next=next;

}

}

class DC\_LinkedList

{

constructor()

{

this.start=null; //i m creating null linked list this.size=0;

}

create(n)

{

var i=1,ec=0,nm=" ",s=0.0;

Page | 9

|  |  |
| --- | --- |
|  | let oldnode=null; let newnode=null; do |
|  | {  ec=parseInt(prompt("Enter Emp Code ")); nm=prompt("Enter Name "); s=parseFloat(prompt("Enter Salary ")); newnode=new DENode(ec,nm,s); |
|  | if(newnode==null) |
|  | { document.write("overflow<br />"); |
|  | alert("overflow"); |
| else | } |
|  | { |
|  | if(this.start==null) |
|  | {  this.start=newnode; |
| else | } |
|  | {  oldnode.next=newnode; |
|  | newnode.prev=oldnode; |
|  | } |
|  | oldnode=newnode; |
|  | this.size++; i++; |
|  | } |
|  | newnode.next=this.start; this.start.prev=newnode; |
|  | }  while(i<=n); |
|  | } |
|  | insert(ec,nm,s,pos)  { var i=1; |

Page | 10

let newnode=null,ptr=null;

newnode=new DENode(ec,nm,s);

if(pos==1)

{

newnode.next=this.start;

this.start.prev.next=newnode;

newnode.prev=this.start.prev; this.start.prev=newnode;

this.start=newnode;

this.size++;

document.write(":Inserted Node at "+pos+"th position<br />");

} else

{

ptr=this.start;i=1;

while((i<pos-1)&&(ptr.next!=this.start))

{

ptr=ptr.next;

i++;

} if(i==pos-1)

{

newnode.next=ptr.next;newnode.prev=ptr; ptr.next.prev=newnode;

ptr.next=newnode;

this.size++;

document.write(":Inserted Node at "+pos+"th position<br />");

} else

{

alert("wrong position"+pos);

document.write("invalid position<br />");

}

Page | 11

}

}

del(pos)

{

var i=1;

let delnode=null,ptr=null; if(this.start==null)

{

alert("Linklist Underflow");

document.write("linkedlist underflow<br />");

}

else

{

if(pos==1)

{

delnode=this.start;

this.start=this.start.next;

this.start.prev=delnode.prev;

delnode.prev.next=this.start;

document.write("Deleted item="+delnode.code+":"+delnode.name+":"+delnode.sal); document.write(":Deleted "+pos+" th item <br />");

this.size--;

delnode=null;

}

else

{

ptr=this.start;i=1;

while((i<pos-1)&&(ptr.next!=this.start))

{

i++; ptr=ptr.next;

}

if(i==pos-1)

{

delnode=ptr.next;

Page | 12

ptr.next=delnode.next;

delnode.next.prev=ptr;

document.write("Deleted item="+delnode.code+":"+delnode.name+": "+delnode.sal); document.write(":Deleted "+pos+" th item <br />");

this.size--;

delnode=null;

}

else

{

document.write("Invalid Position<br />");

}

}

} }

print()

{

let ptr=this.start; document.write("Linked

List:::");

do

{

document.write(ptr.code+":"+ptr.name+":"+ptr.sal+"::");

ptr=ptr.next;

}

while(ptr!=this.start); document.write("<br/>");

}

}

function dcl()

{

var i=1,ec=0,nm=" ",s=0.0;

Page | 13

const l1=new DC\_LinkedList(); //creating Dll var ch=0,pos=0,n=0;

do

{

ch=parseInt(prompt("1:create 2:insert 3:delete 4:Print 5:exit ...Enter your choice"));

switch(ch)

{

case 1:

n=parseInt(prompt("Enter total nodes to

create")); l1.create(n);break;

case 2:

ec=parseInt(prompt("Enter Emp code ")); nm=prompt("Enter Name "); s=parseFloat(prompt("Enter salary ")); pos=parseInt(prompt("Enter position to

insert")); l1.insert(ec,nm,s,pos);break;

case 3:

pos=parseInt(prompt("Enter position to delete"));

l1.del(pos);break;

case 4: l1.print();break;

case 5:break;

default:alert("Wrong choice");

}document.write(" ---><br />");

}while(ch!=5); } Output:

--- >

Linked List:::1:Shubham

:1000::2:abhi:2000::3:shab:3000::4:prachi:4000::5:adesh:5000::

Page | 14

--->

:Inserted Node at 1th position

--- >

Linked List:::10:shantanu:50000::1:Shubham

:1000::2:abhi:2000::3:shab:3000::4:prachi:4000::5:adesh:5000:: --->

Deleted item=1:Shubham : 1000:Deleted 2 th item

--->

Linked

List:::10:shantanu:50000::2:abhi:2000::3:shab:3000::4:prachi:4000::5 :adesh:5000::

--- >

--->

Page | 15

**3. Write a program to check the parenthesis of expression is well formed ness or not.**

**Solution:**

**/\*HTML Source Code\*/**

<html>

<head>

<script type="text/javascript" src="expr.js" ></script>

</head>

<body>

Click here for the result

<input type="button" onclick="Expr()" value="WelFormedness

Expression" />

</body>

</html>

**/\*JavaScript Source Code\*/**

class Stack {constructor(s)

{this.size=s;

this.stk=new Array(this.size);

this.top=-1;//i m creating empty stact

}

isStackFull()

{if(this.top==this.size-1) return(true);

else

return(false);

}

isStackEmpty()

{if(this.top==-1) return(true);

else

return(false);

}

Page | 16

push(item)

{

if(this.top==this.size-1)

{alert("Stack overflow"); document.write("Stack

Overflow <br/>"); }else

{this.top++; this.stk[this.top]=item;

}

}

pop()

{var item;

if(this.top==-1)

{alert("Stack underflow"); document.write("Stack

underflow <br/>");

}else

{item=this.stk[this.top];

this.top=this.top-1;

}

return(item);

}

}

function Expr() {var i,flag=0; var

str;

str=prompt("Enter Expression : "); alert("Given

Expression :"+str); document.write("Given

String="+str+"<br/>"); const l1=new

Stack(str.length); for(i=0;i<str.length;i++)

{if(str.charAt(i)=="(")

{l1.push(str.charAt(i));

//document.write(str.charAt(i)+" pushed<br />");

}else if(str.charAt(i)==")")

Page | 17

{if(l1.isStackEmpty()==false)

l1.pop();

else

{ document.write("Expression is not

wellFormedness...Extra closing<br/>");

alert("Expression is not

wellFormedness...Extra closing");

flag=1;break;

}

}

}

if(flag==0)

{

if((l1.isStackEmpty()==true)&&(i==str.length))

{ document.write("Expression is

wellFormedness<br/>"); alert("Expression is wellFormedness"); }else

{

document.write("Expression is not

wellFormedness...Extra Openings<br/>"); alert("Expression is not

wellFormedness...Extra

Openings");

}

}

}

/\* Output:

Given String=((X^2)+(Y^2))

Expression is wellFormedness

Output:

Given String=((((a+b)\*c)/(a+b)

Expression is not wellFormedness...Extra Openings

Page | 18

Output:

Given String=(a+b)\*c)

Expression is not wellFormedness...Extra closing \*/

Page | 19

**4. Write a program to create stack using linked list and manipulate it using menu driven code.**

**Solution:**

**/\*HTML Source Code\*/**

<html>

<head>

<script type="text/javascript" src="Stack.js" > </script>

</head>

<body>

Click here for the result

<input type="button" onclick="Stack()" value="Stack" />

</body>

</html>

**/\*JavaScript Source Code\*/**

class SLNode

{

constructor(data,next=null)

{

this.data=data; this.next=next;

}

}

class Stack\_list

{

constructor()

{

this.top=null; //i m creating empty stact

}

push(item)

{

let newnode=null;

newnode=new SLNode(item); if(newnode==null)

Page | 20

{

alert("Stack overflow");

document.write("Stack Overflow <br/>");

}

else

{

if(this.top==null)

{

this.top=newnode;

}

else

{

newnode.next=this.top;

this.top=newnode;

}

document.write(item+"is inserted in the stack");

}

}

pop()

{

let delnode=null; var item;

if(this.top==null)

{

alert("Stack underflow");

document.write("Stack underflow <br/>");

}

else

{

item=this.top.data;delnode=this.top;

this.top=this.top.next; delnode=null; alert("Deleted item"+item);

document.write("deleted item "+item+"<br/>");

}

}

printStack()

Page | 21

{

alert("in print"); let ptr=this.top;

document.write("Stack ::"); while(ptr)

{

alert("in print"+ptr.data);

document.write(ptr.data+":");

ptr=ptr.next;

}

document.write(":<br />");

}

}

function Stack()

{

const l1=new Stack\_list(); //creating stack var ch=0,item=0;

do

{

ch=parseInt(prompt("1:Push 2:Pop 3:Print stack 4:exit ...Enter your choice"));

switch(ch)

{case 1: item=parseInt(prompt("Enter item to push")); l1.push(item);break; case 2:

l1.pop();break; case 3:l1.printStack();break; case 4:break;

default:alert("Wrong choice");

}

document.write(" ---><br/>");

}

while(ch!=4); }

OUTPUT:

10is inserted in the stack --->

20is inserted in the stack --->

30is inserted in the stack --->

Page | 22

40is inserted in the stack ---> Stack ::40:30:20:10::

--- >

deleted item 40

--- >

deleted item 30

--- >

Stack ::20:10::

--- >

--->

Page | 23

**5. Write a program to create CQ using array and manipulate it using menu driven code.**

**Solution:**

**/\*HTML Source Code\*/**

<html>

<head>

<script type="text/javascript" src="Array\_CQ.js" ></script>

</head>

<body>

Click here for the result

<input type="button" onclick="CQ()" value="Queue" />

</body>

</html>

**/\*JavaScript Source Code\*/**class CQueue {constructor(s)

{

this.size=s;

this.Q=new Array(this.size);

this.front=-1; this.rear=-1;

}

insertQ(data)

{ if(((this.front==0)&&(this.rear==this.size-

1)) ||(this.rear==(this.front-1)))

{document.write("<Br/>Q overflow"); alert("Q overflow");

}else

{this.rear++;if(this.rear==this.size)this.rear=0; this.Q[this.rear]=data; if(this.front==-

1)this.front++;

document.write("<Br/>"+data+"is inserted

successfully in Q");

Page | 24

alert(data+"is inserted successfully in Q");

}

}

delQ()

{if((this.front==-1)&&(this.rear==-1))

{document.write("<Br/>Q underflow");

alert("Q underflow");

}else

{var data=this.Q[this.front]; if(this.front==this.rear)

{this.front=-1;

this.rear=-1;

}else

{

this.front++;

if(this.front==this.size)this.front=0;

}

document.write("<Br/>"+data+"is deleted

successfully from Q");

alert(data+"is deleted successfully from Q");

return(data);

}

}

printQ()

{var i=0,qstr="Q::"; document.write("<Br/>Q::"); if((this.front==-1)&&(this.rear==-1))return;

if(this.front<=this.rear)

{for(i=this.front;i<=this.rear;i++)

{document.write(this.Q[i]+":");

qstr=qstr+this.Q[i]+":"

}

}else

{for(i=this.front;i<this.size;i++)

{document.write(this.Q[i]+":");

Page | 25

qstr=qstr+this.Q[i]+":"

}

for(i=0;i<=this.rear;i++)

{document.write(this.Q[i]+":");

qstr=qstr+this.Q[i]+":"

}

}

alert(qstr);

}

}

function CQ()

{ var q=new CQueue(5); var ch=0,data=0; for(;;)

{ ch=parseInt(prompt("1:InsertQ 2:DeleteQ 3:PrintQ 4:Exit Enter choice"));

switch(ch)

{

case 1:data=parseInt(prompt("Enter Data"));

q.insertQ(data);break;

case 2:data=q.delQ();break;

case 3:q.printQ();

}

if(ch==4)break;

}

}

/\* Output:

11is inserted successfully in Q

22is inserted successfully in Q

33is inserted successfully in Q Q::11:22:33:

44is inserted successfully in Q

55is inserted successfully in Q Q::11:22:33:44:55: Q overflow

Page | 26

11is deleted successfully from Q Q::22:33:44:55:

22is deleted successfully from Q

33is deleted successfully from Q 77is inserted successfully in Q Q::44:55:77: 44is deleted successfully from Q Q::55:77:

55is deleted successfully from Q Q::77:

77is deleted successfully from Q

Q::

Q::\*/

Page | 27

**6. Write a program to reverse stack using queue.**

**Solution:**

**/\*HTML Source Code\*/**

<html>

<head>

<script type="text/javascript" src="ReverseStack.js" > </script>

</head>

<body>

Click here for the result

<input type="button" onclick="ReverseStack()" value="Reverse Stack" />

</body>

</html>

**/\*JavaScript Source Code\*/**

class SLNode

{

constructor(data,next=null)

{

this.data=data; this.next=next;

}

}

class DQ

{

constructor()

{

this.front=null;this.rear=null//i m creating empty Q

}

insertQ(item)

{

let newnode=null;

newnode=new SLNode(item); if(newnode==null)

{

alert("Q overflow");

Page | 28

document.write("Q Overflow <br/>");

}

else

{

if((this.rear==null)&&(this.front==null))

{

this.rear=newnode;this.front=newnode;

}

else

{

this.rear.next=newnode;

this.rear=newnode;

}

document.write(item+" is inserted in the Q<br/>");

}

}

delQ()

{

let delnode=null;

var item;

if((this.front==null)&&(this.rear==null))

{

alert("Q underflow");

document.write("Q underflow <br/>");

}

else

{

item=this.front.data;

delnode=this.front;

if(this.front==this.rear)

{

this.front=null;this.rear=null;

}

else

{

Page | 29

this.front=this.front.next;

delnode=null;

alert("Deleted item : "+item);

}

document.write("deleted item : "+item+"<br/>");

}

return(item);

}

printQ()

{

var qt="";

let ptr=this.front; document.write("Q ::"); while(ptr)

{ qt=qt+ptr.data+":";

document.write(ptr.data+":"); ptr=ptr.next;

}

document.write(":<br />");alert("Q::"+qt);

}

}

function Q()

{

const l1=new DQ();//creating dynamic Q var ch=0,item=0;

do

{

ch=parseInt(prompt("1:Insert 2:Delete 3:Print Q 4:exit ...Enter your choice")); switch(ch)

{

case 1: item=parseInt(prompt("Enter item to push"));

l1.insertQ(item);break; case 2:

l1.delQ();break; case 3: l1.printQ();break;

case 4:break;

default:alert("Wrong choice");

Page | 30

}

document.write(" ---><br/>");

}while(ch!=4);

}

class Stack\_list

{ constructor()

{

this.top=null; //i m creating empty stact

}

push(item)

{

let newnode=null;

newnode=new SLNode(item); if(newnode==null)

{

alert("Stack overflow");

document.write("Stack Overflow <br/>");

}

else

{

if(this.top==null)

{ this.top=newnode;

}

else

{

newnode.next=this.top;

this.top=newnode;

}

document.write(item+"is inserted in the stack<br/>");

}

}

pop()

{

let delnode=null; var item;

Page | 31

if(this.top==null)

{

alert("Stack underflow");

document.write("Stack underflow <br/>");

}

else

{

item=this.top.data;delnode=this.top;

this.top=this.top.next; delnode=null; alert("Deleted

item: "+item);

document.write("deleted item: "+item+"<br/>");

}

return(item);

}

printStack()

{ var stk="";

let ptr=this.top; document.write("Stack ::");

while(ptr)

{

stk=stk+ptr.data+":";

document.write(ptr.data+":");

ptr=ptr.next;

}

document.write(":<br />");

alert("Stack ::"+stk);

}

}

function Stack()

{

const l1=new Stack\_list();//creating stack var ch=0,item=0;

do

{

ch=parseInt(prompt("1:Push 2:Pop 3:Print stack 4:exit ...Enter your choice")); switch(ch)

Page | 32

{

case 1: item=parseInt(prompt("Enter item to push")); l1.push(item);break; case 2:

l1.pop();break; case 3:l1.printStack();break; case 4:break;

default:alert("Wrong choice");

}

document.write(" ---><br/>");

}

while(ch!=4);

}

function ReverseStack()

{

const stk=new Stack\_list(); var n=0,i=0,item=0; n=parseInt(prompt("Enter Total elements of the stack"));

for(i=0;i<n;i++)

{

item=parseInt(prompt("Enter item to push"));

stk.push(item);

}

stk.printStack(); const q=new DQ();

while(stk.top!=null)

{

item=stk.pop();

q.insertQ(item);

}

while(!((q.rear==null)&&(q.front==null)))

{

item=q.delQ(); stk.push(item);

}

stk.printStack();

}

OUTPUT:

10is inserted in the stack

20is inserted in the stack

Page | 33

30is inserted in the stack

40is inserted in the stack

50is inserted in the stack Stack ::50:40:30:20:10:: deleted item: 50 50 is inserted in the Q deleted item: 40 40 is inserted in the Q deleted item: 30 30 is inserted in the Q deleted item: 20 20 is inserted in the Q deleted item: 10 10 is inserted in the Q deleted item : 50 50is inserted in the stack deleted item : 40 40is inserted in the stack deleted item : 30 30is inserted in the stack deleted item : 20 20is inserted in the stack deleted item : 10 10is inserted in the stack Stack ::10:20:30:40:50::

Page | 34

**7. Write a program to create Binary search tree and traverse it using recursive preorder, inorder, postorder methods.**

**Solution:**

**/\*HTML Source Code\*/**

<html>

<head>

<script type="text/javascript" src="BST\_DFT\_BFT.js" ></script>

</head>

<body>

Click here for the result

<input type="button" onclick="Tree()" value="BST using DFT and BFT" />

</body>

</html>

**/\*JavaScript Source Code\*/**class SLNode

{constructor(data,next=null)

{

this.data=data;

this.next=next;

}

}

class Stack\_list

{constructor()

{this.top=null;//i m creating empty stact

}

push(item)

{let newnode=null; newnode=new

SLNode(item);

if(newnode==null)

{alert("Stack overflow"); document.write("Stack Overflow <br/>"); }else

{

if(this.top==null)

{

this.top=newnode;

}else

{newnode.next=this.top; this.top=newnode;

}

//document.write(item+"is inserted in the stack<br/>");

}

}

pop()

{let delnode=null;var item;

if(this.top==null)

{alert("Stack underflow"); document.write("Stack

underflow <br/>"); }else

{item=this.top.data;delnode=this.top; this.top=this.top.next; delnode=null;

//alert("Deleted item: "+item);

//document.write("deleted item: "+item+"<br/>");

}

return(item);

}

printStack()

{var stk="";

let ptr=this.top; document.write("Stack ::");

while(ptr)

{stk=stk+ptr.data+":"; document.write(ptr.data+":");

ptr=ptr.next;

}document.write(":<br />");

alert("Stack ::"+stk);

}

}

class DQ {constructor()

{this.front=null;this.rear=null//i m creating empty Q

}

insertQ(item)

{let newnode=null;

newnode=new SLNode(item);

if(newnode==null)

{alert("Q overflow"); document.write("Q

Overflow <br/>");

}else

{

if((this.rear==null)&&(this.front==null))

{

this.rear=newnode;this.front=newnode;

}else

{this.rear.next=newnode; this.rear=newnode;

}

//document.write(item+" is inserted in the

Q<br/>");

}

}

delQ()

{

let delnode=null;var item; if((this.front==null)&&(this.rear==null))

{alert("Q underflow"); document.write("Q

underflow <br/>");

}else

{

item=this.front.data;delnode=this.front;

if(this.front==this.rear)

{ this.front=null;this.rear=null; }else this.front=this.front.next; delnode=null;

//alert("Deleted item : "+item);

//document.write("deleted item : "+item+"<br/>");

}

return(item);

}

printQ()

{

var qt="";

let ptr=this.front;

document.write("Q ::");

while(ptr)

{qt=qt+ptr.data+":";

document.write(ptr.data+":");

ptr=ptr.next;

}

document.write(":<br />");alert("Q::"+qt);

}

}

class treeNode

{constructor(data,left=null,right=null) {

this.left=left

this.data=data;

this.right=right;

}

}

class BST {constructor()

{this.root=null;//i m creating empty Tree

}

is\_BST\_Empty()

{

Page | 38

if(this.root==null) return true;

else

return false;

}

is\_BST\_Full()

{let newnode=null; newnode=new treeNode(100);

if(newnode==null)

return true;

else

{newnode=null;return false;

}

}

create(n)

{var i=1;let newnode=null,temp=null;var item=0;

for(i=1;i<=n;i++)

{item=parseInt(prompt("Enter item:")); newnode=new treeNode(item);

if(this.root==null)

{

this.root=newnode;

}else

{temp=this.root; do

{if(newnode.data<=temp.data) { if(temp.left==null)

{ temp.left=newnode;break; }else

temp=temp.left;

}else

{if(temp.right==null)

{ temp.right=newnode;break; }else

temp=temp.right;

Page | 39

}

}while(temp!=null);

}

}

}

DFT()

{const stk=new Stack\_list(); document.write("<br/>DFT::");

let temp=this.root;

do

{while(temp!=null)

{document.write(temp.data+":");

if(temp.right!=null)stk.push(temp.right);

temp=temp.left;

}

if(stk.top!=null)temp=stk.pop();

}while(temp!=null);

}

BFT()

{const q=new DQ();

document.write("<br/>BFT::");

let temp=this.root;q.insertQ(this.root); while((q.rear!=null)&&(q.front!=null)) {

temp=q.delQ();

document.write(temp.data+":");

if(temp.left!=null)q.insertQ(temp.left);

if(temp.right!=null)q.insertQ(temp.right);

}

}

}

function Tree()

{

const t1=new BST();

if(t1.is\_BST\_Empty()) document.write("Null Tree");

else document.write("<Br/>Not Null Tree");

t1.create(5);

if(t1.is\_BST\_Empty()) document.write("Null Tree"); else document.write("<Br/>Not Null Tree");

t1.DFT(); t1.BFT();

}

/\*

Output:

Preorder::34:23:67:56:89:

Inorder::23:34:56:67:89:

Postorder::23:56:89:67:34:

\*/

Page | 41

**8. Write a program to implement graph using adjacency matrix and traverse it using BFT and DFT.**

**Solution:**

**/\*HTML Source Code\*/**

<html>

<head>

<script type="text/javascript" src="Graph\_BFT1.js" >

</script>

</head>

<body>

Click here for the result

<input type="button" onclick="Graph()" value="Graph Traversal

Using BFT" />

</body>

</html>

**/\*JavaScript Source Code\*/** class SLNode

{constructor(data,next=null)

{

this.data=data; this.next=next;

}

}

class DQ {constructor()

{this.front=null;this.rear=null//i m creating empty Q

}

insertQ(item)

{let newnode=null; newnode=new

SLNode(item);

if(newnode==null)

{alert("Q overflow"); document.write("Q Overflow <br/>");

}else

{

if((this.rear==null)&&(this.front==null))

{

this.rear=newnode;this.front=newnode;

}else

{this.rear.next=newnode; this.rear=newnode;

}

//document.write(item+" is inserted in the

Q<br/>");

}

}

delQ()

{

let delnode=null;var item; if((this.front==null)&&(this.rear==null))

{alert("Q underflow"); document.write("Q

underflow <br/>");

}else

{

item=this.front.data;delnode=this.front;

if(this.front==this.rear)

{ this.front=null;this.rear=null; }else this.front=this.front.next; delnode=null;

//alert("Deleted item : "+item);

//document.write("deleted item : "+item+"<br/>");

}

return(item);

}

printQ()

{

var qt="";

let ptr=this.front;

Page | 43

document.write("Q ::");

while(ptr)

{qt=qt+ptr.data+":"; document.write(ptr.data+":");

ptr=ptr.next;

}

document.write(":<br />");alert("Q::"+qt);

}

}

function BFT(n,gfg)

{alert("in bft"); var i=0,v=0,w=0;

const q=new DQ();//creating dynamic Q

var flag=new Array(n); for (i = 0; i < n; i++)

{

flag[i] = 0; }v=0;//source

node

q.insertQ(v);alert("V"+(v+1)+":");

flag[v]=1;

while(1)

{v=q.delQ();document.write("V"+(v+1)+":");

for(w=0;w<n;w++)

{if((gfg[v][w]==1)&&(flag[w]==0))

{

q.insertQ(w);

//document.write("V"+(w+1)+":");

flag[w]=1;//alert("V"+(w+1)+":");

}

}

if((q.front==null)&&(q.rear==null))break;

}

}

function Graph()

{

var n=0,i=0,j=0; n=parseInt(prompt("Enter total vertex of the graph"));

var gfg = new Array(n); for (i = 0; i < gfg.length; i++)

{ gfg[i] = new Array(n);

}

document.write("Now start to enter graph <br>");

for(i=0;i<n;i++)

{

for(j=0;j<n;j++)

{

if(i!=j)

{

gfg[i][j]=parseInt(prompt("Enter edge for

vertex V"+(i+1)+" to V"+(j+1)+": 1: edge exist 0:edge not exist")); }else gfg[i][j]=0;

}

}

document.write("Adjecent Matrix of Graph<br>");

for (var i = 0; i < n; i++)

{

for (var j = 0; j < n; j++)

{

document.write(gfg[i][j] + " ");

}

document.write("<br>");

}

BFT(n,gfg);

}

/\*

Output:

Now start to enter graph

Adjecent Matrix of Graph

Page | 45

01100

10110

11001

01001

00110 V1:V2:V3:V4:V5:

\*/

Page | 46

**9. Write a program to implement linear probing hashing for insert and search operation.**

**Solution:**

**/\*HTML Source Code\*/**

<html>

<head>

<script type="text/javascript" src="Hash\_LB.js" > </script>

</head>

<body>

Click here for the result

<input type="button" onclick="Hashing()" value="Hash Table with Linear Probing" />

</body>

</html>

**/\*JavaScript Source Code\*/**class class HashTable

{constructor(s)

{this.hashTable=new Array(s);//i m creating HashTable

}

init\_Hash()

{for(var i=0;i<this.hashTable.length;i++)

this.hashTable[i]=" ";

}

print\_Hash()

{var altr=""; document.write("<Br/>Hash

Table<Br/>"); for(var

i=0;i<this.hashTable.length;i++)

{

if(this.hashTable[i]!=" ")

{

document.write("<Br/>"+i+":"+this.hashTable[i]); altr=altr+this.hashTable[i]+":";

Page | 47

}

}alert(altr);

}

hashFunction(item)

{return(item.charCodeAt(0));

}

insert(item)

{var i=this.hashFunction(item);var j=i;

if(this.hashTable[i]==" ")

{

this.hashTable[i]=item; document.write("<Br/>

"+item+"is succesfully

inserted in Hash table");

alert("This "+item+"is succesfully inserted in Hash table");

}else

{while(this.hashTable[i]!=" ")

{i=(i+1)%this.hashTable.length;if(i==j)break;

}

if(this.hashTable[i]==" ")

{

this.hashTable[i]=item; document.write("<Br/>

"+item+"is succesfully

inserted in Hash table");

alert("This "+item+"is succesfully inserted in

Hash table");

}else if(i==j)

{

document.write("<Br/> Hash Overflow");

alert("Hash Overflow");

}

}

}

search(item)

{var i=this.hashFunction(item);var j=i; do

{

if(this.hashTable[i]==item)

{ document.write("<Br/>"+item+" is found at "+i+"

th location in Hash Table");

alert(item+" is found at "+i+" th location in

Hash Table");return;

}i=(i+1)%this.hashTable.length;if(this.hashTable[i]==" ")break;

}while(i!=j)

if((this.hashTable[i]==" ")||(i==j))

{ document.write("<Br/>"+item+" is not found in Hash

Table");

alert(item+" is not found in Hash Table");

}

}

}

function Hashing()

{const h=new HashTable(128);h.init\_Hash(); var

ch=0,item;

do

{ch=parseInt(prompt("1:Insert 2:Print 3:Search 4:exit

...Enter your choice"));

switch(ch)

{case 1: item=prompt("Enter item to insert");

h.insert(item);break;

case 2: h.print\_Hash();break;

case 3: item=prompt("Enter item to search");

h.search(item);break;

case 4:break;

default:alert("Wrong choice");

}

}while(ch!=4);

Page | 49

}/\*

Output:

111is succesfully inserted in Hash table 222is succesfully inserted in Hash table 333is succesfully inserted in Hash table aaais succesfully inserted in Hash table AAAis succesfully inserted in Hash table

ABCis succesfully inserted in Hash table

BBBBis succesfully inserted in Hash table

112is succesfully inserted in Hash table

Hash Table

49:111

50:222

51:333

52:112

65:AAA

66:ABC

67:BBBB

97: aaa

AA2is succesfully inserted in Hash table

Hash Table

49:111

50:222

51:333

52:112

65:AAA

66:ABC

67:BBBB

68:AA2

97:aaa

\*/

**10. Write a program which implement the rain terraces to measure trapping rain water.**

**Solution:**

**/\*HTML Source Code\*/**

<html>

<head>

<script type="text/javascript" src="TW.js" >

</script>

</head>

<body>

Click here for the result

<input type="button" onclick="TW()"

value="Trapped Water on Rain terraces" />

</body>

</html>

**/\*JavaScript Source Code\*/** function maxWater(e,n)

{var stw=0,i=0,left=0,right=0,j=0,min;

for(i=1;i<n-1;i++)

{

left=e[i];

for(j=0;j<i;j++)

{if(left<e[j])left=e[j];

}

right=e[i];

for(j=i+1;j<n;j++)

{if(right<e[j])right=e[j];

}

if(left<=right)min=left; else min=right;

stw=stw+(min-e[i]);

}

return(stw);

}

function TW()

{alert("Hi"); var el\_map,n,i; n=parseInt(prompt("Enter Array size :"));

el\_map=new Array(n);

for(i=0;i<n;i++)

el\_map[i]=parseInt(prompt("Enter Array element :")); document.write("<Br/>Elivation map

length="+n+"<Br/>Elivation Map ="); for(i=0;i<n;i++)

document.write(el\_map[i]+":"); var tw=maxWater(el\_map,n);

document.write("<Br/> Total Trapped water in above elivation

map="+tw);

}

/\*

Output:

Elivation map length=9

Elivation Map =3:1:2:5:2:3:6:0:7:

Total Trapped water in above elivation map=14 \*/

**11. Write a program to implements to implement Kruskal’s Algorithm MST.**

**Solution:**

**/\*HTML Source Code\*/**

<html>

<head>

<script type="text/javascript" src="K\_MST.js" >

</script>

</head>

<body>

Click here for the result

<input type="button" onclick="graph()" value="Kruskal's Algorithm for minimum spanning tree" />

</body>

</html>

**/\*JavaScript Source Code\*/**class Edge

{constructor(s=0,d=0,w=0) {

//alert("in edge");

this.src=s;

this.dest=d;

this.weight=w;

}

}

class Graph { constructor(v,e)

{//alert("in graph"); this.V=v; this.E=e; //alert("in graph1"); this.Edges=new Array(this.E);

//alert("in graph2");

Page | 53

for(var i=0;i<this.E;i++)

{ this.Edges[i]=new Edge();

}//alert("in graph3");

}

}

class subset { constructor()

{this.parent;

this.rank;

}

}

function find(subsets,i)

{ if(subsets[i].parent!=i)

subsets[i].parent=find(subsets,subsets[i].parent);

return subsets[i].parent;

}

function Union(subsets,x,y)

{var xroot=find(subsets,x); var yroot=find(subsets,y);

if(subsets[xroot].rank<subsets[yroot].rank)

subsets[xroot].parent=yroot;

else if(subsets[xroot].rank>subsets[yroot].rank)

subsets[yroot].parent=xroot;

else

{subsets[yroot].parent=xroot;

subsets[xroot].rank++;

}

}

function sort(g)

{var i,j,temp;

for(i=0;i<g.E;i++) {for(j=i+1;j<g.E;j++)

{if(g.Edges[i].weight>g.Edges[j].weight)

{

temp=g.Edges[i].src;g.Edges[i].src=g.Edges[j].src;g.Edges[j].src=

temp;

temp=g.Edges[i].dest;g.Edges[i].dest=g.Edges[j].dest;g.Edges[j].

dest=temp;

temp=g.Edges[i].weight;g.Edges[i].weight=g.Edges[j].weight;g.E dges[j].weight=temp;

}

}

}

}

function KruskalMST(g)

{alert("In Kruskal"); var

V=g.V; var i,e;

var result=new Array(V);alert("In Kruskal1"+V);

for(i=0;i<V;i++){ result[i]=new Edge(); } alert("In

Kruskal1");

sort(g);

var subsets=new Array(V);

for(i=0;i<V;i++)

{subsets[i]=new subset(); subsets[i].parent=i;

subsets[i].rank=0;

}e=0;i=0; alert("In Kruskal2"); while(e<V-1 && i<g.E)

{var next\_edge=new Edge();

next\_edge=g.Edges[i++];

var x=find(subsets,next\_edge.src); var y=find(subsets,next\_edge.dest);

if(x!=y)

{result[e++]=next\_edge;

Page | 55

Union(subsets,x,y);

}

}

var minCost=0;

document.write("<Br/>Following are the edges in the

constructed MST<Br/>"); for(i=0;i<e;i++)

{ document.write(result[i].src+"--

"+result[i].dest+"=="+result[i].weight+"<Br/>");

minCost=minCost+result[i].weight;

}

document.write("Minimum Cost Spanning Tree

: "+minCost+"<Br/>");

}

function graph()

{alert("Hi");var vt,ed; vt=parseInt(prompt("Enter Total Vertex")); ed=parseInt(prompt("Enter Total Edges")); var i, g=new Graph(vt,ed);

for(i=0;i<g.E;i++)

{g.Edges[i].src=parseInt(prompt("Enter Source"));

g.Edges[i].dest=parseInt(prompt("Enter Destinition"));

g.Edges[i].weight=parseInt(prompt("Enter Weight"));

}

document.write("<Br/>Graph("+g.V+","+g.E+")"); for(i=0;i<g.E;i++)

{ document.write("<Br/>"+g.Edges[i].src+" to

"+g.Edges[i].dest+"weight="+g.Edges[i].weight);

}

KruskalMST(g);

/\*sort(g);

document.write("<Br/>Graph("+g.V+","+g.E+")"); for(i=0;i<g.E;i++)

{ document.write("<Br/>"+g.Edges[i].src+" to

"+g.Edges[i].dest+"weight="+g.Edges[i].weight);

Page | 56

}\*/

}

/\*

Output:

Graph(4,5)

0 to 1weight=10

0 to 2weight=6

1. to 3weight=5
2. to 3weight=15
3. to 3weight=6

Following are the edges in the constructed MST

0--3==5

0--2==6

0--1==10

Minimum Cost Spanning Tree :21

Graph(9,14)

1. to 1weight=4
2. to 2weight=8
3. to 3weight=7
4. to 4weight=9
5. to 5weight=10
6. to 6weight=2
7. to 7weight=1
8. to 0weight=8

1 to 7weight=11

1. to 8weight=7
2. to 2weight=2

8 to 6weight=6

1. to 5weight=4
2. to 5weight=14

Following are the edges in the constructed MST

6--7==1

5--6==2

8--2==2

Page | 57

0--1==4

2--5==4

2--3==7

1--2==8

3--4==9

Minimum Cost Spanning Tree :37

\*/

Page | 58

**12. Write a program to implements to implement Prims Algorithm MST.**

**Solution:**

**/\*HTML Source Code\*/**

<html>

<head>

<script type="text/javascript" src="P\_MST.js" >

</script>

</head>

<body>

Click here for the result

<input type="button" onclick="graph()" value="Prim's Minimum Spanning Tree Algorithm" />

</body>

</html>

**/\*JavaScript Source Code\*/** function minKey(key,mstSet,V)

{var min=66000,min\_index,v; for(v=0;v<V;v++)

{if(mstSet[v]==false && key[v]<=min)

{min=key[v];min\_index=v;

}

}

return min\_index;

}

function primMST(graph,src,v)

{alert(" in primMST"); var key=new Array(v); var

parent=new Array(v);

var mstSet=new Array(v),i;alert(" in primMST1");

for(i=0;i<v;i++)

{

key[i]=66000;mstSet[i]=false;

Page | 59

}

key[0]=src;

parent[src]=-1;

var count=0,n=0,u=0;alert(" in primMST2"); for(count=0;count<v-1;count++) { u=minKey(key,mstSet,v); mstSet[u]=true;

for(n=0;n<v;n++)

{

if( !mstSet[n] && graph[u][n] &&

graph[u][n]<key[n])

{parent[n]=u;key[n]=graph[u][n];

}

}

}

document.write("<Br/>Vertex:::Distance From Source Node

0<BR/>"); for(i=1;i<v;i++)

{ document.write(parent[i]+"--

"+i+":::"+graph[i][parent[i]]+"<Br/>");

}

}

function graph()

{ alert("Hi"); var

v,g,i,j;

v=parseInt(prompt("Enter total number of vertices :")); g=new Array(v);

for(i=0;i<v;i++) g[i]=new

Array(v); for(i=0;i<v;i++)

{for(j=0;j<v;j++) { g[i][j]=-1;

}

Page | 60

}

for(i=0;i<v;i++)

{for(j=0;j<v;j++)

{ if(i==j) {

g[i][j]=0;

}else

{if(g[i][j]==-1)

{

g[i][j]=parseInt(prompt("Enter distance of node "+i+" to node "+j+" :"));

g[j][i]=g[i][j];

}

}

}

}

document.write("<Br/>Total Vertyex="+v+"<Br/>Graph

=<Br/>");

for(i=0;i<v;i++)

{for(j=0;j<v;j++)

{

document.write(" "+g[i][j]);

}

document.write("<Br/>");

}

primMST(g,0,v);

}

/\*

Output:

Total Vertyex=9

Graph =

040000080

4080000110

080704002

Page | 61

0070914000

0009010000

00414100200

000002016

8110000107

002000670

Vertex:::Distance From Source Node 0

0--1:::4

5--2:::4

1. --3:::7
2. --4:::9

6--5:::2

7--6:::1

0--7:::8

2--8:::2

Total Vertex=5

Graph =

02060

20385

03007

68009

05790

Vertex:::Distance From Source Node 0

0--1:::2

1--2:::3

0--3:::6

1--4:::5

Page | 62

Page | 63

**13. Write a program to implement binary search algorithm using divide and conquer technique.**

**Solution:**

**/\*HTML Source Code\*/**

<html>

<head>

<script type="text/javascript" src="BS.js" ></script>

</head>

<body>

Click here for the result

<input type="button" onclick="BS()" value="Binary Search" />

</body>

</html>

**/\*JavaScript Source Code\*/**function binarySearch(A,LB,UB,it)

{var midPoint; alert("LB="+LB+"

UB="+UB); if(UB>=LB)

{midPoint=parseInt(LB+(UB-LB)/2);

if(A[midPoint]==it)

return(midPoint);

else if(A[midPoint]<it)

{ return(binarySearch(A,midPoint+1,UB,it)); }else

{return(binarySearch(A,LB,midPoint-1,it));

}

}

return(-1);

}

function BS()

{ alert("hi"); var i,item,loc=-1,n=parseInt(prompt("Enter total elements")); var num=new Array(n);

Page | 64

for(i=0;i<n;i++)

{num[i]=parseInt(prompt("Enter element"));

}

for(i=0;i<n;i++)

{document.write("<Br/>"+num[i]);

}

item=parseInt(prompt("Enter element which you want to

search"));

document.write("<Br/>item="+item); loc=binarySearch(num,0,n-1,item);

if(loc!=-1)

{ document.write("<Br/>"+item+" is found at location in num

"+loc);

}else

{document.write("<Br/>"+item+" is not found in num");

}

}

/\*

OUTPUT:

11

22

33

44

55

66

77

88

99

122 item=88

88 is found at location in num 7\*/

**14. Write a program to implement TOWER OF HANNOI problem.**

**Solution:**

**/\*HTML Source Code\*/**

<html>

<head>

<script type="text/javascript" src="TOH.js" ></script>

</head>

<body>

Click here for the result

<input type="button" onclick="mainTower()" value="Tower of

Hannoi" />

</body>

</html>

**/\*JavaScript Source Code\*/** function

TOH(disk,source,dest,aux)

{alert("in TOH disk="+disk);

if(disk==1)

{ document.write("<Br/> Move Disk from Tower

"+source+" to Tower "+dest);

}else

{TOH(disk-1,source,aux,dest); document.write("<Br/>

Move Disk from Tower

"+source+" to Tower "+dest); TOH(disk-

1,aux,dest,source);

}

}

function mainTower()

{alert("hi"); var

n;

n=parseInt(prompt("Enter total number of the disk"));

document.write("<Br/>Total disk="+n); TOH(n,"A","C","B");

}

/\* Output:

Total disk=4

Move Disk from Tower A to Tower B Move Disk from Tower A to Tower C Move Disk from Tower B to Tower C

Move Disk from Tower A to Tower B Move Disk from Tower C to Tower A Move Disk from Tower C to Tower B

Move Disk from Tower A to Tower B Move Disk from Tower A to Tower C Move Disk from Tower B to Tower C Move Disk from Tower B to Tower A

Move Disk from Tower C to Tower A Move Disk from Tower B to Tower C

Move Disk from Tower A to Tower B Move Disk from Tower A to Tower C

Move Disk from Tower B to Tower C \*/

**15. Write a program to find longest common sub sequence.**

**Solution:**

**/\*HTML Source Code\*/**

<html>

<head>

<script type="text/javascript" src="LCSubSq.js" >

</script>

</head>

<body>

Click here for the result

<input type="button" onclick="LCS()" value="Longest Common Sub Sequence" />

</body>

</html>

**/\*JavaScript Source Code\*/**function Print\_LCS(B,X,i,j)

{

if((i==0)&&(j==0))

{//document.write(X.charAt(i)); return;

}

if(B[i][j]=="D")

{Print\_LCS(B,X,i-1,j-1);

document.write(X.charAt(i-1));//alert(x.charAt(i));

}else if(B[i][j]=="U")

Print\_LCS(B,X,i-1,j); else

Print\_LCS(B,X,i,j-1);

}

function longestCommonSubSequence(x,m,y,n)

{var c=new Array(m+1); var b=new Array(m+1);

var i,j,L=0; for(i=0;i<m+1;i++)

{c[i]=new Array(n+1); b[i]=new Array(n+1);

}

for(i=0;i<m+1;i++){c[i][0]=0;b[i][0]="x";}

for(j=0;j<n+1;j++){c[0][j]=0;b[0][j]="x";} for(i=1;i<=m;i++)

{for(j=1;j<=n;j++) {

b[i][j]="x";

if(x.charAt(i-1)==y.charAt(j-1)) {

c[i][j]=c[i-1][j-1]+1;

if(L<c[i][j])L=c[i][j];

b[i][j]="D";

}else if(c[i-1][j]>=c[i][j-1])

{

c[i][j]=c[i-1][j];

b[i][j]="U";

}else

{c[i][j]=c[i][j-1];

b[i][j]="L";

}

}

}

document.write("<BR/>Matrix C<BR/>");

for(i=0;i<=m;i++)

{//alert("this is "+i);

for(j=0;j<=n;j++)

{//alert("this is "+j+"::"+c[i][j]+":"+b[i][j]);

document.write(c[i][j]+":"+b[i][j]+" ");

}document.write("<Br/>");

}

document.write("<Br/>Longest Common subsequence = "); Print\_LCS(b,x,m,n);

return(c[m][n]);

}

function LCS()

{alert("Hi"); var x,y,m,n;

x=prompt("Enter First sub sequence");

y=prompt("Enter second sub sequence"); m=x.length; n=y.length;

document.write("<Br/>Sequence 1 = "+x+" length =

"+m+"<Br/>Sequence 2 = "+y+" length="+n);

len=longestCommonSubSequence(x,m,y,n); document.write("<Br/> Length of LCSubsequence="+len);

}

/\*

Output:

Sequence 1 = GXTXYAB length = 7

Sequence 2 = AGGTAB length=6

Matrix C

0:x 0:x 0:x 0:x 0:x 0:x 0:x

0:x 0:U 1:D 1:D 1:L 1:L 1:L

0:x 0:U 1:U 1:U 1:U 1:U 1:U

0:x 0:U 1:U 1:U 2:D 2:L 2:L

0:x 0:U 1:U 1:U 2:U 2:U 2:U

0:x 0:U 1:U 1:U 2:U 2:U 2:U

0:x 1:D 1:U 1:U 2:U 3:D 3:L

0:x 1:U 1:U 1:U 2:U 3:U 4:D

Longest Common subsequence = GTAB

Length of LCSubsequence=4 \*/

**16. Write a program to implement regular expression to check whether given pattern match with input string or not?**

**Solution:**

**/\*HTML Source Code\*/**

<html>

<head>

<script type="text/javascript" src="RE.js" ></script>

</head>

<body>

Click here for the result

<input type="button" onclick="RE()" value="Regular Expression" />

</body>

</html>

**/\*JavaScript Source Code\*/**function strMatchPattern(str,ptrn)

{var n=str.length,m=ptrn.length;

if(m==0) return (n==0);

var i,j, lookup=new Array(n+1);

for(i=0;i<=n;i++) {lookup[i]=new Array(m+1);

}

for(i=0;i<=n;i++)

{for(j=0;j<=m;j++) lookup[i][j]=false;

}

lookup[0][0]=true;

//\*match

for(j=1;j<=m;j++)

if(ptrn.charAt(j-1) == "\*") lookup[0][j]=lookup[0][j-1];

//fill table..bottom up fashion for(i=1;i<=n;i++)

{

for(j=1;j<=m;j++)

{

if(ptrn.charAt(j-1)=='\*')

{ lookup[i][j]=lookup[i][j-1] || lookup[i-1][j]; }else

if(ptrn.charAt(j-1) == "?" || str.charAt(i-

1)== ptrn.charAt(j-1))

{ lookup[i][j]=lookup[i-1][j-1]; }else

lookup[i][j]=false;

}

}

return lookup[n][m];

}

function RE()

{alert("Hi"); var x,y; x=prompt("Enter String:"); y=prompt("Enter Pattern:");

document.write("<Br/>String = "+x+" length =

"+x.length+"<Br/>Pattern = "+y+" length="+y.length);

if(strMatchPattern(x,y))

document.write("<Br/> Yes");

else

document.write("<Br/> No");

}

/\*

Output:

String = babbac length = 6

Pattern = b\*?ac length=5

Yes

String = bghfdsmn length = 8

Page | 72

Pattern = bg\*mn length=5

Yes

String = asdf length = 4

Pattern = as?b length=4

No

\*/

Page | 73

**17. Write a program to implement N queen Problem using backtracking approach.**

**Solution:**

**/\*HTML Source Code\*/**

<html>

<head>

<script type="text/javascript" src="NQueen.js" ></script>

</head>

<body>

Click here for the result

<input type="button" onclick="Queen()" value="N Queen Problem"

/>

</body>

</html>

**/\*JavaScript Source Code\*/**function printBoard(b)

{ document.write("<Br/>Queens Placed at following position on the board<Br/>");

for(var i=0;i<b.length;i++)

{document.write("<Br/>::::");

for(var j=0;j<b.length;j++)

document.write(b[i][j]+"::::");

}

}

function printBoard1(b)

{ document.write("<Br/><Br/>Queens Placed at following

position on the board<Br/>"); for(var i=0;i<b.length;i++)

{document.write("<Br/>"); for(var j=0;j<b.length;j++)

document.write(b[i][j]);

}

Page | 74

}

function isSafe(b,r,c,N)

{var i,j;

for(i=0;i<c;i++)

if(b[r][i]==1)

return false;

for(i=r,j=c;j>=0 && i>=0; i--,j--)

if(b[i][j]==1)

return false;

for(i=r,j=c;j>=0 && i<N; i++,j--)

if(b[i][j]==1)

return false;

return true;

}

function placeNQueen(b,c,N)

{if(c>=N)return true; for(var i=0;i<N;i++)

{if(isSafe(b,i,c,N))

{

b[i][c]=1;

if(placeNQueen(b,c+1,N))

return true;

b[i][c]=0;

}

}

return false;

}

function Queen()

{ //alert("Hi"); var n=parseInt(prompt("Enter total number of queens")); var board=new Array(n);

var i,j;

for(i=0;i<n;i++)

{board[i]=new Array(n);

}

Page | 75

for(i=0;i<n;i++)

{for(j=0;j<n;j++)

{board[i][j]=0;

}

}

if(placeNQueen(board,0,n)==false)

{document.write("<Br/> Solution does not exist"); return;

}

printBoard(board);

printBoard1(board);

}

/\*

Output:

Queens Placed at following position on the board:

::::0::::0::::0::::1::::0::::0::::

::::1::::0::::0::::0::::0::::0::::

::::0::::0::::0::::0::::1::::0:::: ::::0::::1::::0::::0::::0::::0:::: ::::0::::0::::0::::0::::0::::1::::

::::0::::0::::1::::0::::0::::0::::

Output 2:

Queens Placed at following position on the board

::::1::::0::::0::::0::::0::::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::::0::::1::::

Page | 76

::::0::::1::::0::::0::::0::::0::::0::::0:::: ::::0::::0::::0::::1::::0::::0::::0::::0:::: ::::0::::0::::0::::0::::0::::1::::0::::0:::: ::::0::::0::::1::::0::::0::::0::::0::::0::::

Output 3:

Queens Placed at following position on the board:

::::1::::0::::0::::0::::0::::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::::0::::1::::

::::0::::1::::0::::0::::0::::0::::0::::0:::: ::::0::::0::::0::::1::::0::::0::::0::::0:::: ::::0::::0::::0::::0::::0::::1::::0::::0::::

::::0::::0::::1::::0::::0::::0::::0::::0::::

\*/

Page | 77