

B.Sc III Semester
Data Structures using C++ Practical Solutions

Not as per Syllabus Sheet-only a Few Important

1. Write a program to Implement Stack Using Array

```
#include<iostream.h>
#include<conio.h>
class Stack
{
    int top, a[5];
public:
    Stack()
    {   top=-1;   }
    void push(int ele);
    int pop();
};
void Stack::push(int ele)
{
    if(top>=4)
        cout<<"stack is full";
    else
        a[++top]=ele;
}
int Stack::pop()
{
    int d;
    if(top<0)
    {
        cout<<"stack is empty";
        return 0;
    }
    else
    {
        d=a[top--];
        return d;
    }
}
void main()
{
    Stack s;
    int i;
    clrscr();
    s.push(10);        s.push(20);
    s.push(30);        s.push(40);
    s.push(50);
    cout<<"Stack Elements are:"<<"\n";
    i=1;
```

```

while(i<=5)
{
    cout<<" "<<s.pop();
    i++;
}
    getch();
}

```

Output:

**Stack Elements are
50 40 30 20 10**

2. Write a program to convert to POSTFIX expression?

```
#include<iostream.h>
```

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

```
int prec(char c)
```

```
{
    if(c == '^')
        return 3;
    else if(c == '*' || c == '/')
        return 2;
    else if(c == '+' || c == '-')
        return 1;
    else
        return -1;
}

```

```
void infixToPostfix(string s)
```

```
{
    std::stack<char> st;
    st.push('N');
    int l = s.length();
    string ns;
    for(int i = 0; i < l; i++)
    {
        if((s[i] >= 'a' && s[i] <= 'z') ||
           (s[i] >= 'A' && s[i] <= 'Z'))
            ns+=s[i];
        else if(s[i] == '(')
            st.push('(');
        else if(s[i] == ')')
        {
            while(st.top() != 'N' && st.top() != '(')
            {
                char c = st.top();
                st.pop();
            }
            ns += c;
        }
    }
}

```

```

        if(st.top() == '(')
        {
            char c = st.top();
            st.pop();
        }
    }
    else
    {
        while(st.top() != 'N' && prec(s[i]) <=prec(st.top()))
        {
            char c = st.top();
            st.pop();
            ns += c;
        }
        st.push(s[i]);
    }
}
while(st.top() != 'N')
{
    char c = st.top();
    st.pop();
    ns += c;
}
cout << ns << endl;
}
void main()
{
    string exp = "a+b*(c^d-e)^(f+g*h)-i";
    infixToPostfix(exp);
    return 0;
}

```

output:

abcd^e-fgh*+^*+i-

3. Write a program to solve Tower of Hanoi Problem

```

#include <iostream.h>
void towerOfHanoi(int n, char A, char B, char C)
{
    if (n == 1)
    {
        cout << "Move disk 1 from rod " << A <<" to rod " << B<<endl;
        return;
    }
    towerOfHanoi(n - 1, A, C, B);
    cout << "Move disk " << n <<" frm rod " <<A<<" to rod" <<B<< endl;
}

```

```
    towerOfHanoi(n - 1, C, B, A);
}
int main()
{
    int n = 4;
    towerOfHanoi(n, 'A', 'C', 'B');
    return 0;
}
```

OUTPUT:

```
Move Disk 1 from rod A to rod C
Move Disk 2 from rod A to rod B
Move Disk 1 from rod C to rod B
Move Disk 3 from rod A to rod C
Move Disk 1 from rod B to rod A
Move Disk 2 from rod B to rod C
Move Disk 1 from rod A to rod C
```

4. Write a program to implement QUEUE operations using array?

```
#include<iostream.h>
#include<conio.h>
#define size 10
class Queue
{
    int a[size];
    int rear;
    int front;
public:
    Queue()
    {
        rear=front=-1;
    }
    void enQueue(int ele);
    int deQueue();
    void display();
};
void Queue::enQueue(int ele)
{
    if(front==size-1)
        front++;
    if(rear==size-1)
        cout<<"Queue is Full";
    else
        a[++rear]=ele;
}

int Queue::deQueue()
{

```

```
        if(front>rear)
        {
            rear=front=-1;
            cout<<"Queue is empty";
            return 0;
        }
        else
            return a[front++];
    }

void Queue::display()
{
    int i;
    for(i=front;i<=rear;i++)
        cout<<a[i]<<"\t";
}

void main()
{
    Queue q;
    clrscr();
    q.enqueue(10);
    q.enqueue(20);
    q.enqueue(30);
    q.enqueue(40);
    q.enqueue(50);
    cout<<"Queue elements are:"<<"\n";
    q.display();
    cout<<"\n Deleted element from the Queue:"<<q.dequeue();
    cout<<"\n Deleted element from the Queue:"<<q.dequeue();
    cout<<"\n After deleting two elements from Queue :"<<"\n";
    q.display();
    getch();
}
```

OUTPUT:

Queue elements are:

10 20 30 40 50

Deleted element from the Queue:10

Deleted element from the Queue:20

After deleting two elements from Queue :

30 40 50

5. Write a program to demonstrate Circular QUEUE?

```
#include<iostream.h>
```

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

```
class Cqueue
```

```
{
```

```
    private:
```

```
int Rear, Front;
int Queue[50];
int Max,Size;
public:
Cqueue()
{
    Size=0;
    Max=50;
    Rear=Front=-1;
}
int Empty();
int Full();
void Add(int Element);
int Delete();
int getFront();
};
int Cqueue::Empty()
{
    if(Front==Rear)
        return 1;
    else
        return 0;
}
int Cqueue::Full()
{
    if(Rear==Front)
        return 1;
    else
        return 0;
}
void Cqueue::Add(int Element)
{
    if(!Full())
        Rear=(Rear+1)%Max;
        Queue[Rear]=Element;
        Size++;
}
int Cqueue::Delete()
{
    if(!Empty())
        Front=(Front+1)%Max;
        Size--;
        return(Queue[Front]);
}
int Cqueue::getFront()
{
    int Temp;
```

```
        if(!Empty())
            Temp=(Front+1)%Max;
        return (Queue[Temp]);
    }
void main()
{
    clrscr();
    Cqueue Q;
    Q.Add(2);
    Q.Add(4);
    Q.Add(6);
    cout<<Q.Delete()<<endl;
    Q.Add(8);
    cout<<Q.Delete()<<endl;
    cout<<Q.Delete()<<endl;
    cout<<Q.Delete()<<endl;
    Q.Add(3);
    Q.Add(5);
    cout<<Q.Delete()<<endl;
    getch();
}
```

OUTPUT:

```
6
8
4
2
5
```

6. Write a program for implementation of Priority Queue?

```
#include<iostream.h>
#include<conio.h>
int a[5],n=0;
void insert(int v)
{
    int i;
    if(n==0)
    { a[n++]=v; }
    else
    {
        for(i=n-1;i>=0;i--)
        {
            if(v>a[i])
            {
                a[i+1]=a[i];
            }
        }
    }
}
```

```

        else
        {
            break;
        }
    }
    a[i+1]=v;
    n++;
}
}
int Delete()
{    return a[--n];    }
void show()
{
    cout<<"Priority Queue Element"<<endl;
    for(int i=0;i<n;i++)
        cout<<"\t"<<a[i];
}

void main()
{
    clrscr();
    insert(30);
    insert(20);
    insert(50);
    insert(10);
    show();
    Delete();
    cout<<"\nAfter Deleting"<<endl;
    show();
    getch();
}

```

Output:

Priority Queue Element

50 30 20 10

After Deleting

Priority Queue Element

50 30 20

7. Write a program to demonstrate DEQUEUE?

```

#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
#define SIZE 10
class dequeue
{
    int a[20],f,r;

```



```
public:
    dequeue();
    void insert_at_beg(int);
    void insert_at_end(int);
    void delete_fr_front();
    void delete_fr_rear();
    void show();
};
dequeue::dequeue()
{
    f=-1;
    r=-1;
}
void dequeue::insert_at_end(int i)
{
    if(r>=SIZE-1)
    {
        cout<<"\nInsertion is not possible, overflow!";
    }
    else
    {
        if(f== -1)
        {
            f++;
            r++;
        }
        else
        {
            r=r+1;
        }
        a[r]=i;
        cout<<"\n Inserted item is:"<<a[r];
    }
}
void dequeue::insert_at_beg(int i)
{
    if(f== -1)
    {
        f=0;
        a[++r]=i;
        cout<<"\nInserted element is:"<<i;
    }
    else if(f!=0)
    {
        a[--f]=i;
        cout<<"\nInserted element is:"<<i;
    }
    else
    {
        cout<<"\nInsertion is not possible, overflow!";
    }
}
```

```
void dequeue::delete_fr_front()
{
    if(f==-1)
    {
        cout<<"Deletion is not possible::dequeue is empty";
        return;
    }
    else
    {
        cout<<"The deleted element is:"<<a[f];
        if(f==r)
        {
            f=r--1;
            return;
        }
        else
            f=f+1;
    }
}
void dequeue::delete_fr_rear()
{
    if(f==-1)
    {
        cout<<"Deletion is not possible::Dequeue is empty";
        return;
    }
    else
    {
        cout<<"The deleted element is:"<<a[r];

        if(f==r)
        {
            f=r--1;
        }
        else
            r=r-1;
    }
}
void dequeue::show()
{
    if(f==-1)
    {
        cout<<"Dequeue is empty";
    }
    else
    {
        for(int i=f;i<=r;i++)
        {
            cout<<a[i]<<" ";
        }
    }
}
```

```
}
int main()
{
    int c,i;
    clrscr();
    dequeue d;
    do
    {
        cout<<"\n 1. Insert at beginning";
        cout<<"\n 2. Insert at end";
        cout<<"\n 3. Show";
        cout<<"\n 4. Deletion from Front";
        cout<<"\n 5. Deletion from Rear";
        cout<<"\n 6. Exit";
        cout<<"\n Enter your Choice:";
        cin>>c;
        switch(c)
        {
            case 1: cout<<"Enter the element to be inserted:";
                cin>>i;
                d.insert_at_beg(i);
                break;
            case 2: cout<<"\n Enter the Element to be insert:";
                cin>>i;
                d.insert_at_end(i);
                break;
            case 3: d.show();
                break;
            case 4: d.delete_fr_front();
                break;
            case 5: d.delete_fr_rear();
                break;
            case 6: exit(0);
            default: cout<<"invalid choice";
                break;
        }
    }while(c!=7);
    return
    getch();
}
```

OUTPUT:

1. Insert at beginning
2. Insert at end
3. Show
4. Deletion from Front
5. Deletion from Rear
6. Exit

Enter your Choice:1
Enter the element to be inserted:10
Inserted element is:10
1. Insert at beginning
2. Insert at end
3. Show
4. Deletion from Front
5. Deletion from Rear
6. Exit

Enter your Choice:2
Enter the element to be inserted:20
Inserted element is:20
1. Insert at beginning
2. Insert at end
3. Show
4. Deletion from Front
5. Deletion from Rear
6. Exit

Enter your Choice:3
10 20
1. Insert at beginning
2. Insert at end
3. Show
4. Deletion from Front
5. Deletion from Rear
6. Exit

Enter your Choice:5
The deleted element is:20
1. Insert at beginning
2. Insert at end
3. Show
4. Deletion from Front
5. Deletion from Rear
6. Exit

Enter your Choice:6

8. Write a program for Insertion and deletion of a node using Single Linked List?

```
#include<iostream.h>
#include<conio.h>
struct Node
{
    int data;
    Node *next;
};
struct Node *head=NULL;
class SingleList
{
```

```
public:
void addNode(int n)
{
struct Node *temp=new Node;
if(head==NULL)
{
temp->data=n;
temp->next=NULL;
head=temp;
}
else
{
struct Node *tmp;
tmp=head;
while(tmp->next!=NULL)
{ tmp=tmp->next; }
temp->data=n;
temp->next=NULL;
tmp->next=temp;
}
}

void display()
{
if(head==NULL)
{
cout<<"List is empty";
}
else
{
cout<<"Linked list elements : "<<endl;
struct Node *temp;
temp=head;
while(temp!=NULL)
{
cout<<temp->data<<" ";
temp=temp->next;
} }
}

void deleteItem()
{
if(head==NULL)
{
cout<<"List is empty"<<endl;
}
struct Node *temp;
temp=head;
```

```
        head=head->next;
        cout<<"\n"<<temp->data<<" is deleted"<<endl;
        temp->next=NULL;
    }
};
void main()
{
    SingleList l;
    clrscr();
    l.addNode(10);
    l.addNode(20);
    l.addNode(30);
    l.addNode(40);
    l.addNode(50);
    l.display();
    l.deleteItem();
    l.display();
    getch();
}
```

Output:

```
Linked list elements :
10 20 30 40 50
10 is deleted
Linked list elements :
20 30 40 50
```

9. Write a program for implementation of Stack using Linked List?

```
#include<iostream.h>
#include<conio.h>
struct Node
{
    int data;
    Node *next;
};
struct Node *head=NULL;
class SingleList
{
public:
void addNode(int n)
{
    struct Node *temp=new Node;
    if(head==NULL)
    {
        temp->data=n;
        temp->next=NULL;
        head=temp;
    }
}
```

```
    else
    {
        struct Node *tmp;
        tmp=head;
        while(tmp->next!=NULL)
        {
            tmp=tmp->next;
        }
        tmp->data=n;
        tmp->next=NULL;
        tmp->next=tmp;
    }
}
void display()
{
    if(head==NULL)
    {
        cout<<"List is empty";
    }
    else
    {
        cout<<"Linked list elements :"<<endl;
        struct Node *temp;
        temp=head;
        while(temp!=NULL)
        {
            cout<<temp->data<<" ";
            temp=temp->next;
        }
    }
}

void deleteItem()
{
    if(head==NULL)
    {
        cout<<"List is empty"<<endl;
    }
    struct Node *temp;
    temp=head;
    head=head->next;
    cout<<"\n"<<temp->data<<" is deleted"<<endl;
    temp->next=NULL;
}
};
void main()
{
```

```
SingleList l;  
clrscr();  
l.addNode(10);  
l.addNode(20);  
l.addNode(30);  
l.addNode(40);  
l.addNode(50);  
l.display();  
l.deleteItem();  
l.display();  
getch();  
}
```

OUTPUT:

```
Linked list elements :  
10 20 30 40 50  
10 is deleted  
Linked list elements :  
20 30 40 50
```

10. Write a program for Queue implementation using Linked List?

```
#include<iostream.h>  
#include<conio.h>  
struct Node  
{  
    int data;  
    Node *next;  
};  
struct Node *rear=NULL;  
struct Node *front=NULL;  
class QueueList  
{  
    public:  
    void insert(int val)  
    {  
        if(rear==NULL)  
        {  
            rear=new Node;  
            rear->data=val;  
            rear->next=NULL;  
            front=rear;  
        }  
        else  
        {  
            struct Node *temp=new Node;  
            temp->data=val;  
            temp->next=NULL;  
            rear->next=temp;  
        }  
    }  
};
```



```
        rear=temp;
    }
}
void deleteElement()
{
    struct Node *temp;
    temp=front;
    if(front==NULL)
        cout<<"Queue is empty";
    else if(temp->next!=NULL)
    {
        temp=temp->next;
        cout<<"\nElement deleted from Queue is:"<<front->data<<endl;
        front->next=NULL;
        front=temp;
    }
    else
    {
        cout<<"\nElement deleted from Queue is:"<<front->data<<endl;
        front=rear=NULL;
    }
}
void display()
{
    struct Node *ptr;
    ptr=front;
    if((front==NULL)&&(rear==NULL))
    {   cout<<"Queue is empty"<<endl;   }
    else
    {
        cout<<"Queue elements are:";
        while(ptr!=NULL)
        {
            cout<<ptr->data<<" ";
            ptr=ptr->next;
        }
    }
}
};
void main()
{
    QueueList q;
    clrscr();
    q.insert(10);
    q.insert(20);
    q.insert(30);
    q.insert(40);
```

```
    q.insert(50);
    q.display();
    q.deleteElement();
    q.display();
    getch();
}
```

OUTPUT:

Queue elements are:

10 20 30 40 50

Element deleted from Queue is:10

Queue elements are:

20 30 40 50

11. Write a program for Tree Traversal in Binary Tree?

```
#include<iostream.h>
```

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

```
struct Tree
```

```
{
```

```
    int data;
```

```
    Tree *left;
```

```
    Tree *right;
```

```
};
```

```
struct Tree *t=NULL;
```

```
class Btree
```

```
{
```

```
    public:
```

```
    void insert(int item)
```

```
    {
```

```
        t=insert(t,item);
```

```
    }
```

```
    Tree* insert(Tree *t,int item)
```

```
    {
```

```
        if(t==NULL)
```

```
        {
```

```
            t=new Tree;
```

```
            t->data=item;
```

```
            t->left=NULL;
```

```
            t->right=NULL;
```

```
        }
```

```
        else if(item<t->data)
```

```
        {
```

```
            t->left=insert(t->left,item);
```

```
        }
```

```
        else if(item>t->data)
```

```
        {            t->right=insert(t->right,item); }
```

```
    return t;
}

void LVR(Tree *t)
{
    if(t!=NULL)
    {
        LVR(t->left);
        cout<<" "<<t->data;
        LVR(t->right);
    }
}

void VLR(Tree *t)
{
    if(t!=NULL)
    {
        cout<<" "<<t->data;
        VLR(t->left);
        VLR(t->right);
    }
}

void LRV(Tree *t)
{
    if(t!=NULL)
    {
        LRV(t->left);
        LRV(t->right);
        cout<<" "<<t->data;
    }
}

void inorder()
{
    cout<<"\n";
    cout<<"Inorder Traversal"<<"\n";
    LVR(t);
}

void preorder()
{
    cout<<"\n";
    cout<<"preorder Traversal"<<"\n";
    VLR(t);
}

void postorder()
{

```

```
    cout<<"\n";
    cout<<"postorder Traversal"<<"\n";
    LRV(t);
}
};
void main()
{
    Btree bt;
    clrscr();
    bt.insert(45);
    bt.insert(35);
    bt.insert(65);
    bt.insert(25);
    bt.insert(40);
    bt.insert(85);
    bt.insert(55);
    bt.inorder();
    bt.preorder();
    bt.postorder();
    getch();
}
```

OUTPUT:

```
Inorder traversal
25 35 40 45 55 65 85
Preorder traversal
45 35 25 40 65 55 85
Postorder traversal
25 40 35 55 85 65 45
```

12. Write a program to demonstrate Linear Search?

```
#include<iostream.h>
#include<conio.h>
int search(int arr[], int n, int x)
{
    int i;
    for (i = 0; i < n; i++)
        if (arr[i] == x)
            return i;
    return -1;
}

int main()
{
    int arr[] = { 3, 4, 1, 7, 5 };
    int n = sizeof(arr) / sizeof(arr[0]);
    int x;
```

```
cout<<"Enter Element To Search";
cin>>x;
int index = search(arr, n, x);
if (index == -1)
    cout << "Element is not present in the array";
else
    cout << "Element found at position " << index;

return 0;
}
```

OUTPUT:

Enter Element To Search 1
Element found at position 2

13. Write a program for implementation of Binary Search?

```
#include<iostream.h>
#include<conio.h>
int bSearch(int[], int, int, int);
int main()
{
    int num[10] = {10, 22, 37, 55, 92, 118};
    int search_num, loc=-1;
    cout<<"Enter the number that you want to search: ";
    cin>>search_num;
    loc = bSearch(num, 0, 6, search_num);
    if(loc != -1)
    {
        cout<<search_num<<" found in the array at the location: "<<loc;
    }
    else
    {
        cout<<"Element not found";
    }
    return 0;
}
int bSearch(int a[], int low, int high, int search_num)
{
    int middle;
    if(high >= low)
    {
        middle = (low + high)/2;
        if(a[middle] == search_num)
        {
            return middle+1;
        }
        else if(a[middle] < search_num)
        {
```

```
        return bSearch(a,middle+1,high,search_num);
    }
    else
    {
        return bSearch(a,low,middle-1,search_num);
    }
}
return -1;
}
```

OUTPUT:

Enter the number that you want to search:55

55 found in the array at the location:3

14. Write a program to sort using Selection Sort?

```
#include<iostream.h>
#include<conio.h>
class SelectSort
{
public:
void selectionSort(int a[],int size)
{
int lowest,pos,i,j;
for(i=0;i<size-1;i++)
{
lowest=a[i];
pos=i;
for(j=i+1;j<size;j++)
{
if(a[j]<lowest)
{
lowest=a[j];
pos=j;
}
}
a[pos]=a[i];
a[i]=lowest;
}
}
void printArray(int a[],int size)
{
int i;
for(i=0;i<size;i++)
{
cout<<a[i]<<" ";
}
cout<<endl;
}
};
```

```
void main()
{
int arr[]={9,2,7,5,4};
SelectSort s;
clrscr();
cout<<"Before Sorting:"<<endl;
s.printArray(arr,5);
s.selectionSort(arr,5);
cout<<"After Sorting:"<<endl;
s.printArray(arr,5);
getch();
}
```

Output:

Before Sorting 9 2 7 5 4
After Sorting 2 4 5 7 9

15. Write a program to implement Bubble sort?

```
#include<iostream.h>
#include<conio.h>
class BSort
{
public:
void bubbleSort(int a[],int size)
{
int t,i,j;
for(i=0;i<size-1;i++)
{
for(j=0;j<size-i-1;j++)
{
if(a[j]>a[j+1])
{
t=a[j]; a[j]=a[j+1]; a[j+1]=t;
}
}
}
}
void printArray(int a[],int size)
{
int i;
for(i=0;i<size;i++)
{
cout<<a[i]<<" ";
}
cout<<endl;
}
};

void main()
{
int arr[]={7,9,2,4,5};
BSort b;
clrscr();
```

```
cout<<"Before Sorting array elements:"<<endl;
b.printArray(arr,5);
cout<<"After Sorting array elements:"<<endl;
b.bubbleSort(arr,5);
b.printArray(arr,5);
getch();
}
```

Output:

Before Sorting array elements:

7 9 2 4 5

After Sorting array elements:

2 4 5 7 9

16. Write a program for Insertion Sort?

```
#include <iostream.h>
#include <conio.h>
void insertionSort(int arr[], int n)
{
    int i, key, j;
    for (i = 1; i < n; i++)
    {
        key = arr[i];
        j = i - 1;
        while (j >= 0 && arr[j] > key)
        {
            arr[j + 1] = arr[j];
            j = j - 1;
        }
        arr[j + 1] = key;
    }
}
void printArray(int arr[], int n)
{
    int i;
    for (i = 0; i < n; i++)
        cout << arr[i] << " ";
    cout << endl;
}
int main()
{
    int arr[] = { 12, 11, 13, 5, 6 };
    int n = sizeof(arr) / sizeof(arr[0]);
    insertionSort(arr, n);
    printArray(arr, n);
    return 0;
}
```

OUTPUT: 5 6 11 12 13

17. Write a program to implement Quick Sort?

```
#include<iostream.h>
#include<conio.h>
class Quick
{
public:
int partition(int a[],int l,int r)
{
int i,j,pivot,t;
pivot=a[l];
i=l;
j=r;

do
{
do
{
i=i+1;
if(i==r) break;
}while(a[i]<=pivot);

do
{
j=j-1;
}while(a[j]>pivot);
if(i<j)
{
t=a[i];
a[i]=a[j];
a[j]=t;
}
}while(i<=j);
a[l]=a[j];
a[j]=pivot;
return j;
}
void qSort(int a[],int l,int r)
{
int j;
if(l<r)
{
j=partition(a,l,r+1);
qSort(a,l,j-1);
qSort(a,j+1,r);
}
}
```

```
void printArray(int a[],int size)
{
    int i;
    for(i=0;i<size;i++)
    {
        cout<<a[i]<<" ";
    }
    cout<<endl;
}
};
void main()
{
    int arr[]={8,1,5,14,4,15,12,6,2,11,10,7,9};
    Quick q;
    clrscr();
    cout<<"Before Sorting array elements:"<<endl;
    q.printArray(arr,13);
    cout<<"After Sorting array elements:"<<endl;
    q.qSort(arr,0,12);
    q.printArray(arr,13);
}
```

Output:

Before Sorting array elements:

8 1 5 14 4 15 12 6 2 11 10 7 9

After Sorting array elements:

1 2 4 5 6 7 8 9 10 11 12 14 15

18. Write a program for Merge Sort?

```
#include<iostream.h>
void swapping(int &a, int &b)
{
    int temp;
    temp = a;
    a = b;
    b = temp;
}
void display(int *array, int size)
{
    for(int i = 0; i<size; i++)
        cout << array[i] << " ";
    cout << endl;
}
void merge(int *array, int l, int m, int r)
{
    int i, j, k, nl, nr;
    nl = m-l+1; nr = r-m;
    int larr[nl], rarr[nr];
```

```
for(i = 0; i<nl; i++)
    larr[i] = array[l+i];
for(j = 0; j<nr; j++)
    rarr[j] = array[m+1+j];
i = 0; j = 0; k = l;
while(i < nl && j<nr)
{
    if(larr[i] <= rarr[j])
    {
        array[k] = larr[i];
        i++;
    }
    else
    {
        array[k] = rarr[j];
        j++;
    }
    k++;
}
while(i<nl)
{
    array[k] = larr[i];
    i++;
    k++;
}
while(j<nr)
{
    array[k] = rarr[j];
    j++;
    k++;
}
}
void mergeSort(int *array, int l, int r)
{
    int m;
    if(l < r)
    {
        int m = l+(r-l)/2;
        mergeSort(array, l, m);
        mergeSort(array, m+1, r);
        merge(array, l, m, r);
    }
}
int main()
{
    int n;
    cout << "Enter the number of elements: ";
```

```

cin >> n;
int arr[n];
cout << "Enter elements:" << endl;
for(int i = 0; i<n; i++)
{
    cin >> arr[i];
}
cout << "Array before Sorting: ";
display(arr, n);
mergeSort(arr, 0, n-1);
cout << "Array after Sorting: ";
display(arr, n);
}

```

19. Write a program to Visit the Nodes of Tree Using BFS Method?

```

#include<iostream>
#include<conio.h>
#include<stdlib.h>
int cost[10][10],i,j,k,n,qu[10],front,rare,v,visit[10],visited[10];
int main()
{
    int m;
    cout <<"Enter no of vertices:";
    cin >> n;
    cout <<"Enter no of edges:";
    cin >> m;
    cout <<"\nEDGES \n";
    for(k=1; k<=m; k++)
    {
        cin >>i>>j;
        cost[i][j]=1;
    }
    cout <<"Enter initial vertex to traverse from:";
    cin >>v;
    cout <<"Visited vertices:";
    cout <<v<<" ";
    visited[v]=1;
    k=1;
    while(k<n)
    {
        for(j=1; j<=n; j++)
            if(cost[v][j]!=0 && visited[j]!=1 && visit[j]!=1)
            {
                visit[j]=1;
                qu[rare++]=j;
            }
        v=qu[front++];
        cout<<v <<" ";
    }
}

```

```

    k++;
    visit[v]=0;
    visited[v]=1;
}
return 0;
}

```

OUTPUT:

Enter Number of vertices:5

Enter no of edges 5

EDGES

1 2

2 3

3 4

1 4

1 5

Enter initial vertex to traverse from:1

Visited vertices:

1 2 4 5 3

20. Write a program to Visit the Nodes of Tree Using DFS Method?

```

#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
int cost[10][10],i,j,k,n,stk[10],top,v,visit[10],visited[10];
int main()
{
    int m;
    cout <<"Enter no of vertices:";
    cin >> n;
    cout <<"Enter no of edges:";
    cin >> m;
    cout <<"\nEDGES \n";
    for(k=1; k<=m; k++)
    {
        cin >>i>>j;
        cost[i][j]=1;
    }
    cout <<"Enter initial vertex to traverse from:";
    cin >>v;

    cout <<"DFS ORDER OF VISITED VERTICES:";
    cout << v <<" ";
    visited[v]=1;
    k=1;
    while(k<n)
    {

```

```
for(j=n; j>=1; j--)  
    if(cost[v][j]!=0 && visited[j]!=1 && visit[j]!=1)  
    {  
        visit[j]=1;  
        stk[top]=j;  
        top++;  
    }  
v=stk[--top];  
cout<<v << " ";  
k++;  
visit[v]=0;  
visited[v]=1;  
}  
return 0;  
}
```

OUTPUT:

Enter Number of vertices:5

Enter no of edges 5

EDGES

1 2

2 3

3 4

1 4

1 5

Enter initial vertex to traverse from:1

DFS order of visited vertices:

1 2 3 4 5