INDEX

Sl. No	DATA STRUCTURE PROGRAMS	Pg. No.
01	Recursion –Tower of Hanoi problem	3
02	Delete and insert elements from an array	5
03	Sequential and binary search :print number of comparison in each case for given datasets	11
04	Insertion sort	16
05	Bubble and selection sort : print number of comparison and exchanges in each case for given data	20
06	Quick sort	25
07	Merge sort	30
08	Conversion in infix expression to postfix	35
09	Evaluation of postfix expression	42
10	Operations in singly linked list	47
11	Create a binary search tree out of given data and traverse it inorder	56
12	Program to merge two linked list	60
13	Program to add and delete elements from a queue	66
14	Operations in circular linked list	71
15	Use a linked stack to reverse a string	80

INDEX

Sl.no	DATABASE MANAGEMENT SYSTEM	Pg.no
1	Create a table students with field sno,sname,sex,mark with sno as primary key and assign suitable constraints for each attribute. Insert five records into the table.	83
2	Create a table department with field ename, salary, dno,dname,place with dno as primary key.	85
3	Create a table emp with fields eno,ename,job,dno,salary with eno as primary key. Insert five records into the table.	87
4	Create a table emp with field with field eno,ename,job,manager,salary,with eno as primary key, Insert five records into the field.	89
5	Create a table department with field dname,dno,salary,designation,ename,place with dno as primary key. Insert five records into the table.	92
6	Create table students with fields sno,sname,mark with sno as primary key and assign suitable constraints for each attribute. Insert five records into the table.	94
7	Create table employee with field eno,ename,job,salary,dept_no . with eno as primary key. Insert five records into the table.	97
8	Create table deposit with acc no,accname,place,bname,amount with accno as primary key. Insert five record into the table.	100
9	Create a table with attribute tid,tname,age,dept,salary,sex with primary key tid. Insert 5 records into the table.	102
10	Create a table customer with field cid,cname,dob,place.	104

1. Recursion –Tower of Hanoi problem

Program

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
void move(int n,char *s,char *i,char *d)
{
if(n>0)
{
move((n-1),s,d,i);
cout<<"disc\t "<<n<<" \t is moved from \t"<<s<<"\tto\t"<<d<<endl;
move((n-1),i,s,d);
}
}
int main()
{
char i,s,d;
int n;
clrscr();
cout<<"\ntower of hanoi\n";
cout<<"enter the number of disc"<<endl;
cin>>n;
move(n,"source","intermediate","destination");
getch();
return 0;
}
```

Output:

```
tower of hanoi
enter the number of disc
3
disc     1       is moved from  source  to      destination
disc     2       is moved from  source  to      intermediate
disc     1       is moved from  destination     to      intermediate
disc     3       is moved from  source  to      destination
disc     1       is moved from  intermediate    to      source
disc     2       is moved from  intermediate    to      destination
disc     1       is moved from  source  to      destination
```

2. Delete and insert elements from an array

```
#include<iostream.h>
#include<conio.h>
#include<process.h>
class array
{
        private:int a[100];
                int n;
        public:void create();
                void insert();
                void del();
                void display();
};
void array::create()
{
        int i;
        cout<<"enter the size\n";
        cin>>n;
        cout<<"enter the elements\n";
        for(i=0;i<n;i++)
                cin>>a[i];
}
void array::insert()
```

```
{
        int num,pos,i;
        cout<<"enter the number\n";
        cin>>num;
        cout<<"enter the position\n";
        cin>>pos;
        if(pos>n)
                cout<<"invalid input\n";
        else
        {
                for(i=n-1;i>=pos-1;i--)
                a[i+1]=a[i];
                a[pos-1]=num;
                n++;
        }
}
void array::del()
{
        int i,pos;
        cout<<"enter the positon\n"<<endl;
        cin>>pos;
        for(i=pos;i<n;i++)
        a[i-1]=a[i];
        n--;
```

```
}
void array::display()
{
        int i;
        for(i=0;i<n;i++)
        cout<<" "<<a[i];
}
void main()
{
        int ch;
        array k;
        clrscr();
        cout<<"\n INSERTION AND DELETION IN ARRAY\N";
        while(1)
        {
                cout<<"\nMENU\n";
                cout<<"\n1.create";
                cout<<"\n2.INSERT";
                cout<<"\n3.DELETE";
                cout<<"\n4.DISPLAY";
                cout<<"\n5.EXIT";
                cout<<"\nenter your choice\n";
                cin>>ch;
                switch(ch)
```

```
            {
                case 1:k.create();
                break;
                case 2:k.insert();
                break;
                case 3:k.del();
                break;
                case 4:k.display();
                break;
                case 5:exit(0);
                default :cout<<"invalid choice";
                break;
            }
        getch();
        }
}
```

Output:

```
 INSERTION AND DELETION IN ARRAYN
MENU

1.create
2.INSERT
3.DELETE
4.DISPLAY
5.EXIT
enter your choice
1
enter the size
4
enter the elements
1 2 3 4

MENU

1.create
2.INSERT
3.DELETE
4.DISPLAY
5.EXIT
enter your choice
4
 1 2 3 4_
```

```
 1 2 3 4
MENU

1.create
2.INSERT
3.DELETE
4.DISPLAY
5.EXIT
enter your choice
2
enter the number
5
enter the position
3

MENU

1.create
2.INSERT
3.DELETE
4.DISPLAY
5.EXIT
enter your choice
4
 1 2 5 3 4
```

```
4
 1 2 5 3 4
MENU

1.create
2.INSERT
3.DELETE
4.DISPLAY
5.EXIT
enter your choice
3
enter the positon

4

MENU

1.create
2.INSERT
3.DELETE
4.DISPLAY
5.EXIT
enter your choice
4
 1 2 5 4
```

3. Sequential and binary search :print number of comparison in each case for given datasets

Program

```
#include<iostream.h>
#include<conio.h>
class array
{
        private:int a[20],count,com;
        public:array();
        void add();
        void binary(int item);
        void linear(int item);
};
array::array()
{
        count=0;
}
void array::add()
{
        int item,size;
        cout<<"enter number of elements\n";
        cin>>size;
        cout<<"enter elements\n";
        for(count=0;count<=size;count++)
```

```
        {
                cin>>item;
                a[count]=item;
        }
}
void array::binary(int item)
{
        int mid,low=0,up=count-1,flag=1;
        for(mid=(low+up)/2;low<=up;mid=(low+up)/2)
        {
                if(a[mid]==item)
                {
                        cout<<"\enter the number is at position"<<mid;
                        flag=0;
                        break;
                }
                if(a[mid]>item)
                {
                        up=mid-1;
                }
                else
                {
                        low=mid+1;
                }
                com++;
```

```
        }
        cout<<"\n no of comparisons in binary search is"<<com;

        if(flag==1)
        cout<<"\n elements is not present in the array";
}
void array::linear(int item)
{
        int flag=0,i=0;
        while(i<count)
        {
                if(a[i]==item)
                {
                        flag=1;
                        cout<<"\n item is found at"<<i<<"th position";
                }
                com++;
                i++;
        }
        cout<<"\n no of comparisons is"<<com;

        if(flag==0)
        {
                cout<<"\n item is not found in the array";
        }
```

```
}
int main()
{
        clrscr();
        int item;
        array b;
        b.add();
        cout<<"\n enter the item to search";
        cin>>item;
        cout<<"\n binary search is";
        b.binary(item);
        cout<<"\nlinear search is";
        b.linear(item);
        getch();
        return 0;
}
```

Output:

```
enter number of elements
4
enter elements
2
4
6
8

 enter the item to search
6

 binary search

 enter the number is at position2
 no of comparison in binary search is1

 linear search

item is found at 2th position
 no of comparison in linear search is5
```

4. Insertion sort

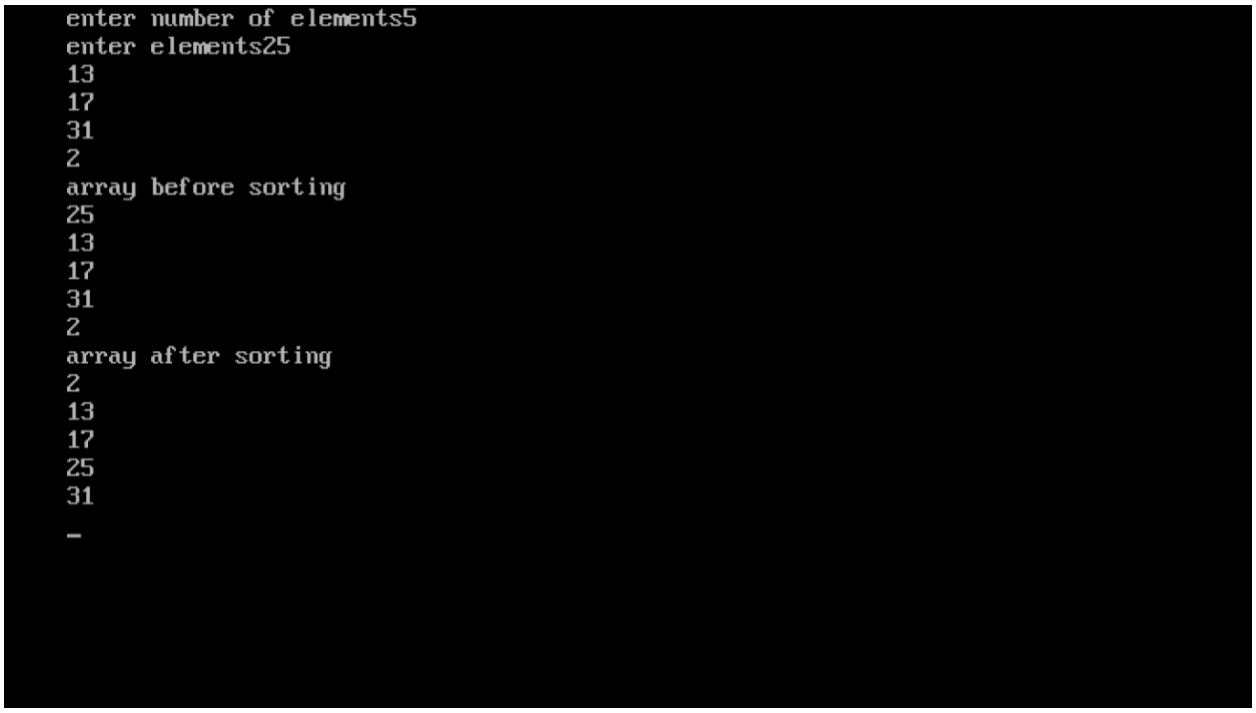
Program

```
#include<iostream.h>
#include<conio.h>
const int max=10;
class array
{
        int a[max],count;
        public:
                array();
                void add();
                void sort();
                void display();
};
array::array()
{
        count=0;
        for(int i=0;i<max;i++)
                a[i]=0;
}
void array::add()
{
        cout<<"enter number of elements";
        cin>>count;
```

```
        cout<<"enter elements";
        for(int i=0;i<count;i++)
        {
                cin>>a[i];
        }
}
void array::sort()
{
        int temp;
        for(int i=1;i<count;i++)
        {
                for(int j=0;j<i;j++)
                {
                        if(a[j]>a[i])
                        {
                                temp=a[j];
                                a[j]=a[i];
                                for(int k=i;k>j;k--)
                                        a[k]=a[k-1];
                                a[k+1]=temp;
                        }
                }
        }
}
void array::display()
```

```
{
        for(int i=0;i<count;i++)
                cout<<a[i]<<"\n";
}
int main()
{
        array a;
        clrscr();
        a.add();
        cout<<"array before sorting\n";
        a.display();
        cout<<"array after sorting\n";
        a.sort();
        a.display();
        getch();
        return 0;
}
```

Output:


enter number of elements5
enter elements25
13
17
31
2
array before sorting
25
13
17
31
2
array after sorting
2
13
17
25
31

5. Bubble and selection sort : print number of comparison and exchanges in each case for given data

Program

```
#include<iostream>
#include<conio.h>
#include<process.h>
class array
{
public:
int arr[100],arr1[100],count;
int j,i;
array();
void add();
void bsort();
void ssort();
};
array::array()
{
for(i=0;i<100;i++)
{
arr[i]=0;
arr1[i]=0;
}
}
void array::add()
```

```
{
cout<<"enter the number of elements"<<endl;
cin>>count;
cout<<"enter the elements"<<endl;
for(i=0;i<count;i++)
{
cin>>arr[i];
}
for(i=0;i<count;i++)
{
        arr1[i]=arr[i];
}
}
void array:: bsort()
{
int temp,c=0,e=0;
for(i=0;i<=count-2;i++)
{
for(j=0;j<=count-2;j++)
{
if(arr1[j]>arr1[j+1])
{
temp=arr1[i];
arr1[j]=arr1[j+1];
arr1[j+1]=temp;
```

```
e++;
}
c++;
}
}
cout<<endl<<"the bubble sorted array is"<<endl;
for(i=0;i<count;i++)
{
cout<<arr1[i]<<"\t";
}
cout<<endl<<"number of comparision"<<c<<endl<<"number of exchanges"<<e<<endl;
}
void array::ssort()
{
 int temp,c=0,e=0;
 for(int i=0;i<=count-2;i++)
 {
 for(int j=i+1;j<=count-1;j++)
 {
 if(arr[i]>arr[j])
 {
 temp=arr[i];
 arr[i]=arr[j];
 arr[j]=temp;
 e++;
```

```
}
c++;
}
}
cout<<endl<<"the selection sorted array is"<<endl;
for(i=0;i<count;i++)
{
cout<<arr[i]<<"\t";
}
cout<<endl<<"the number of comparison"<<c<<endl<<"the number of exchange"<<e;
}
int main()
{
clrscr();
array a;
a.add();
a.ssort();
a.bsort();
getch();
return 0;
}
```

Output:

```
enter the number of elements
5
enter the elements
45
2
34
78
245

the selection sorted array is
2       34      45      78      245
the number of comparison10
the number of exchange2
the bubble sorted array is
2       2       34      78      245
number of comparision16
number of exchanges3
```

6. Quick sort

Program

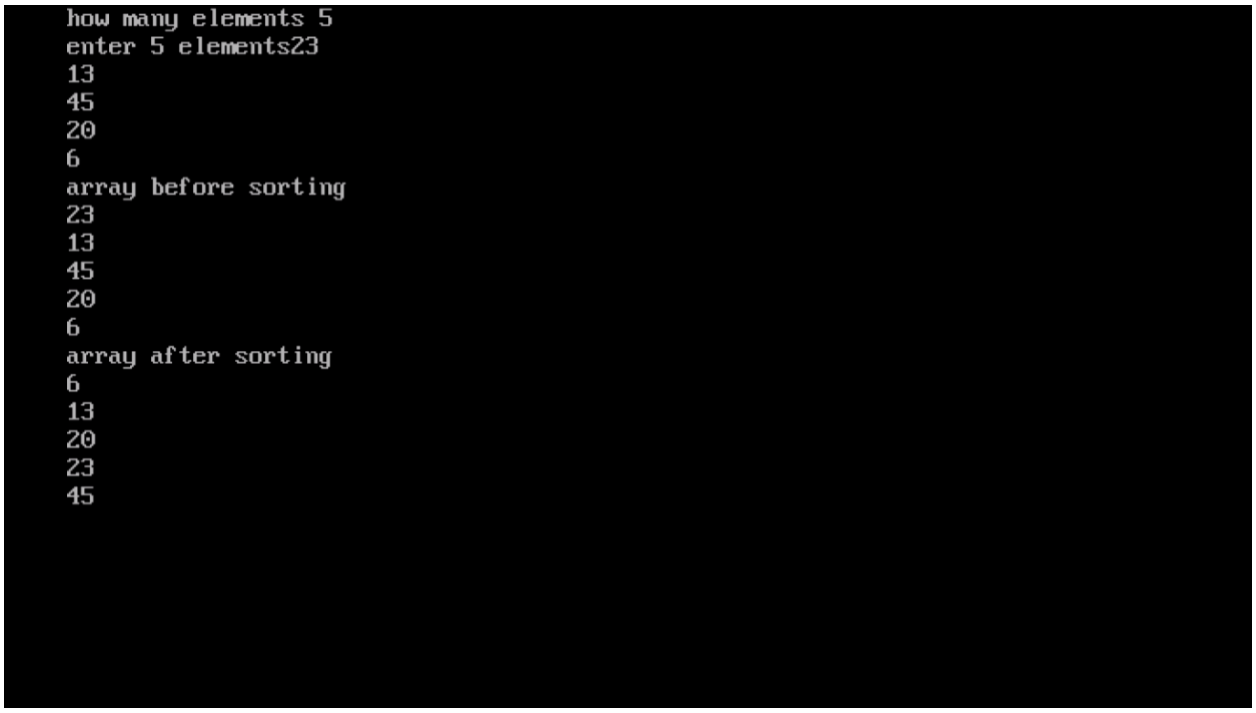
```
#include<iostream.h>
#include<conio.h>
class array
{
        int a[20],count;
        public:array();
        void add();
        int getcount();
        static int split(int *,int,int);
        void qsort(int low,int up);
        void display();
};
array::array()
{
        count=0;
}
void array::add()
{
        int item,size;
        cout<<"how many elements ";
        cin>>size;
        cout<<"enter "<<size<<" elements";
        for(count=0;count<size;count++)
```

```
        {
                cin>>item;
                a[count]=item;
        }
}
int array::getcount()
{
        return count;
}
void array::qsort(int low,int up)
{
        if(up>low)
        {
                int i;
                i=split(a,low,up);
                qsort(low,i-1);
                qsort(i+1,up);
        }
}
int array::split(int *a,int low,int up)
{
        int l,p,q,t;
        p=low+1;
        q=up;
        l=a[low];
```

```
        while(q>=p)
        {
                while(a[p]<l)
                p++;
                while(a[q]>l)
                q--;
                if(q>p)
                {
                        t=a[p];
                        a[p]=a[q];
                        a[q]=t;
                }
        }
        t=a[q];
        a[q]=a[low];
        a[low]=t;
        return q;
}
void array::display()
{
        for(int i=0;i<count;i++)
        {
                cout<<a[i]<<"\n";
        }
}
```

```
int main()
{
        array b;
        clrscr();
        b.add();
        cout<<"array before sorting\n";
        b.display();
        int c=b.getcount();
        b.qsort(0,c-1);
        cout<<"array after sorting";
        b.display();
        getch();
        return 0;
}
```

Output:


how many elements 5
enter 5 elements23
13
45
20
6
array before sorting
23
13
45
20
6
array after sorting
6
13
20
23
45

7. Merge sort

Program

```
#include <iostream.h>
#include <conio.h>
#include <process.h>
 class array{
        int arr[20],size,count;
 public:
        array();
        void add();
        void merge(array a,array b);
        void display();
        static void sort(int *arr,int sz);

 };
 array::array()
 {
        count=size=0;
 }
 void array::add()
 {

        cout<<"enter the number of elements"<<endl;
        cin>>size;
        cout<<"enter the elements"<<endl;
```

```
        for(count=0;count<size;count++)
        {
                cin>>arr[count];
        }
}
void array::merge(array a,array b)
{
        sort(a.arr,a.size);
        sort(b.arr,b.size);
        int i=0,j=0,k=0;
        for(;i<a.count||j<b.count;)
        {
                if(a.arr[i]==b.arr[j])
                {
                        arr[k]=a.arr[i];
                        arr[k+1]=a.arr[i];
                        i++;
                        j++;
                        k=k+2;
                }
                else if(a.arr[i]<=b.arr[j])
                {
                        arr[k]=a.arr[i];
                        i++;
                        k++;
```

```
                }
                else
                {
                        arr[k]=b.arr[j];
                        j++;
                        k++;
                }
        }
        if(i<a.count)
        {
                for(k=i;k<a.count;k++,i++)
                {
                        arr[k]=a.arr[i];
                }
        }
        if(j<b.count)
        {
                for(k=j;k<b.count;k++,j++)
                        arr[k]=b.arr[j];
        }
        count=k;
}
void array::sort(int *arr,int sz)
{
        int temp;
```

```
        for(int i=0;i<=sz-2;i++)
        {
                for(int j=i+1;j<=sz-1;j++)
                {
                        if(arr[i]>arr[j])
                        {
                                temp=arr[i];
                                arr[i]=arr[j];
                                arr[j]=temp;
                        }
                }
        }
}
void array::display()
{
        cout<<"after merge sort";
        for(int i=0;i<count;i++)
        {
                cout<<arr[i]<<"\t";
        }
}
int main()
{
        array a,b,x;
        clrscr();
```

```
        a.add();
        b.add();
        x.merge(a,b);
        x.display();
        getch();
        return 0;
}
```

Output:

```
enter the number of elements
5
enter the elements
678
4
3
36
35
enter the number of elements
4
enter the elements
76
43
2
78
after merge sort2       3       4       35      36      43      76      78      678
```

8. conversion in infix expression to postfix

Program

```
#include<ctype.h>
#include<conio.h>
#include<string.h>
#include<math.h>
#include<iostream.h>



const int max=50;
class infix
{
        char target[max],stack[max];
        char *s,*t;
        int top,v;
        public:infix();
                void setexp(char *str);
                void push(char c);
                char pop();
                void convert();
                int priority(char c);
                void show();
};
```

```
infix::infix()
{
	strcpy(target," ");
	strcpy(stack," ");
	top=0;
	t=target;
	s=" ";
}
void infix::setexp(char*str)
{
	s=str;
}
void infix::push(char c)
{
	if(top==max)
		cout<<"stack is full\n";
	top++;
	stack[top]=c;
}
char infix::pop()
{
	if(top==0)
	{
		cout<<"stack is empty\n";
		return 0;
```

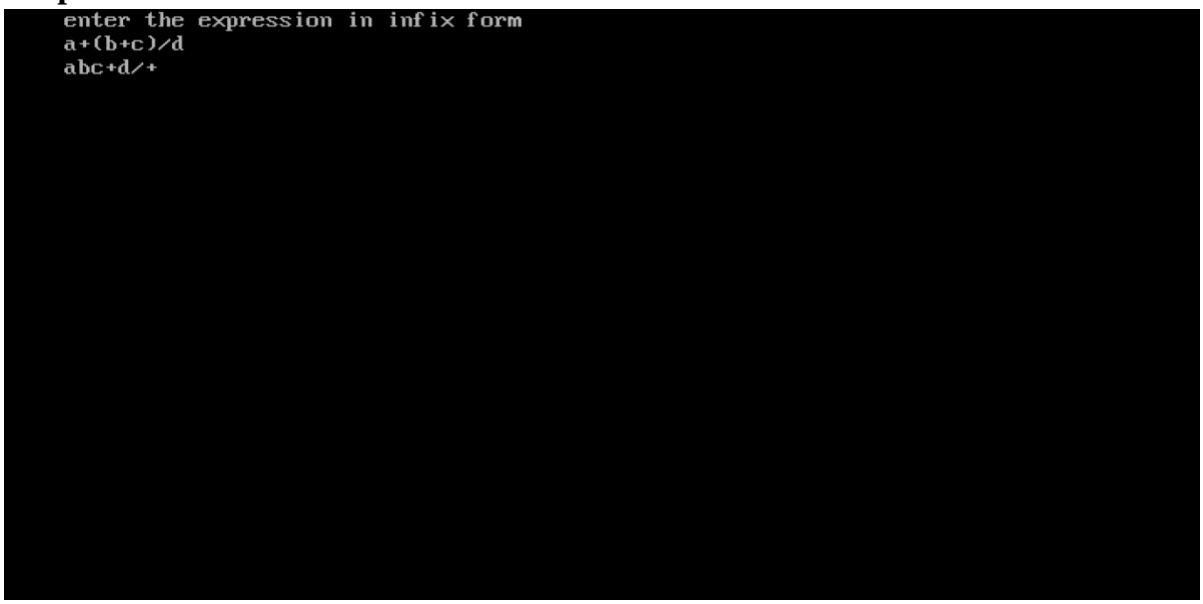
```
        }
        char temp=stack[top];
        top--;
        return temp;
}
void infix::convert()
{
        while(*s)
        {
                if(*s==' '||*s=='\t')
                {
                        s++;
                        continue;
                }
                if(isdigit(*s)||isalpha(*s))
                {
                        while(isdigit(*s)||isalpha(*s))
                        {
                                *t=*s;
                                s++;
                                t++;
                        }
                }
                if(*s=='(')
                {
```

```
            push(*s);
            s++;
}
char opr;
if(*s=='*'||*s=='/'||*s=='+'||*s=='-'||*s=='%'||*s=='$'||*s=='^')
{
            if(top!=0)
            {
                        opr=pop();
                        while((priority(opr))>=(priority(*s)))
                        {
                                    *t=opr;
                                    t++;
                                    opr=pop();
                        }
                        push(opr);
                        push(*s);
            }
            else
            push(*s);
            s++;
}
if(*s==')')
{
            opr=pop();
```

```
                        while((opr)!='(')
                        {
                                *t=opr;
                                t++;
                                opr=pop();
                        }
                        s++;
                }
        }
        while(top!=0)
        {
                char opr=pop();
                *t=opr;
                t++;
        }
        *t='\0';
}
int infix::priority(char c)
{
        if(c=='^'||c=='$')
                return 3;
        if(c=='*'||c=='/')
                return 2;
        if(c=='+'||c=='-')
                return 1;
```

```
        return 0;
}
void infix::show()
{
cout<<target;
}
void main()
{
        char exp[max];
        infix q;
        clrscr();
        cout<<"enter the expression in infix form\n";
        cin.getline(exp,max);
        q.setexp(exp);
        q.convert();
        q.show();
        getch();
}
```

Output:



9. Evalution of postfix expression

Program

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
#include<math.h>
#include<ctype.h>
const int max=50;
class postfix
{
        char*s;
        int stack[max];
        int top,m;
        public:postfix();
            void setexp(char*str);
            void push(int item);
            int pop();
            void calculate();
            void show();
};
postfix::postfix()
{
        top=0;
}
void postfix::setexp(char*str)
```

```
{       s=str;
}
void postfix::push(int item)
{
        if(top==max)
        cout<<"stack is full";
        else
        {
                top++;
                stack[top]=item;
        }
}
int postfix::pop()
{
        if(top==0)
        {
                cout<<"stack is empty\n";
                return 0;
        }
        int data=stack[top];
        top--;
        return data;
}
void postfix::calculate()
{
```

```
int n1,n2,n3;
while(*s)
{
        if(*s==' '||*s=='\t')
        {
                s++;
                continue;
        }
        if(isdigit(*s))
        {
                m=*s-'0';
                push(m);
        }
        else
        {
                n1=pop();
                n2=pop();
                switch(*s)
                {
                        case'+':
                        n3=n2+n1;
                        break;
                        case'-':
                        n3=n2-n1;
                        break;
```

```
                                case'/':n3=n2/n1;
                                break;
                                case'*':n3=n2*n1;
                                break;
                                case'%':n3=n2%n1;
                                break;
                                case'^':n3=pow(n2,n1);
                                break;
                                default:cout<<"unknown expression";
                                break;
                        }
                        push(n3);
                }
                s++;
        }
}
void postfix::show()
{
        m=pop();
        cout<<"result is"<<m;
}
int main()
{
        char exp[max];
        clrscr();
```

```
        cout<<"enter postfix expression\n";
        cin.getline(exp,max);
        postfix q;
        q.setexp(exp);
        q.calculate();
        q.show();
        getch();
        return 0;
}
```

Output:

10. Operations in singly linked list

Program

```
#include<iostream.h>
#include<conio.h>
#include<process.h>
class linklist
{
        struct node
        {
                int data;
                node*link;
        }
        *p;
        public: linklist();
                void append();
                void addatbeg();
                void addafter();
                void display();
                void del();
                ~linklist();
};
linklist::linklist()
{
```

```
        p=NULL;
}
void linklist::append()
{
        node*temp,*r;
        int n;
        cout<<"enter the number\n";
        cin>>n;
        if(p==NULL)
        {
                temp=new node;
                temp->data=n;
                temp->link=NULL;
                p=temp;
        }
        else
        {
                temp=p;
                while(temp->link!=NULL)
                temp=temp->link;
                r=new node;
                r->data=n;
                r->link=NULL;
                temp->link=r;
        }
```

```
}
void linklist::addatbeg()
{
        int n;
        node*temp;
        cout<<"enter the number\n";
        cin>>n;
        temp=new node;
        temp->data=n;
        temp->link=p;
        p=temp;
}
void linklist::addafter()
{
        int n,loc;
        node *temp,*r;
        temp=p;
        cout<<"enter thelocation and no\n";
        cin>>loc>>n;
        for(int i=0;i<loc-1;i++)
        {
                temp=temp->link;
                if(temp==NULL)
                {
                        cout<<"there are less than"<<loc<<"elements list\n";
```

```
                }
        }
        r=new node;
        r->data=n;
        r->link=temp->link;
        temp->link=r;
}
void linklist::display()
{
        node *temp=p;
        cout<<endl;
        while(temp!=NULL)
        {
                cout<<temp->data<<" ";
                temp=temp->link;
        }
}
void linklist::del()
{
        int n;
        node *old,*temp;
        temp=p;
        cout<<"\n enter the number \n";
        cin>>n;
        while(temp!=NULL)
```

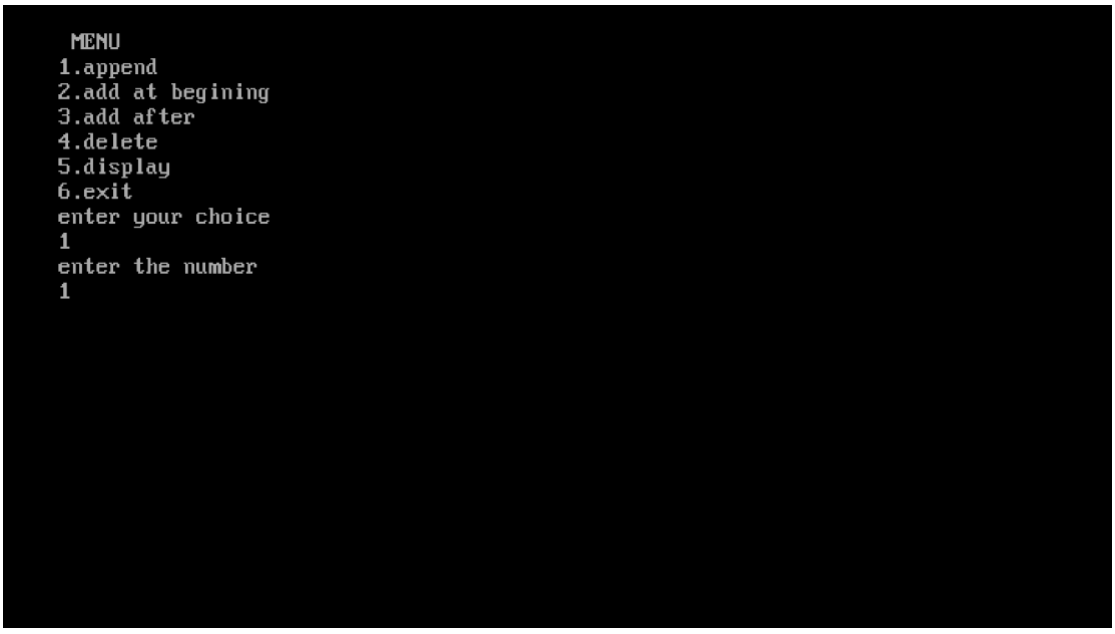
```
        {
                if(temp->data==n)
                {
                        if(temp==p)
                        p=temp->link;
                        else
                        old->link=temp->link;
                        delete temp;
                        return;
                }
                else
                {
                        old=temp;
                        temp=temp->link;
                }
        }
        cout<<"element"<<n<<"not found\n";
}
linklist::~linklist()
{
        node*q;
        while(p!=NULL)
        {
                q=p->link;
                delete p;
```

```
                p=q;
        }
}
void main()
{
        linklist l;
        int ch;
        clrscr();
        do
        {
                cout<<"\n MENU \n";
                cout<<"1.append \n";
                cout<<"2.add at begining \n";
                cout<<"3.add after \n";
                cout<<"4.delete \n";
                cout<<"5.display \n";
                cout<<"6.exit \n";
                cout<<"enter your choice \n";
                cin>>ch;
                switch(ch)
                {
                        case 1:l.append();
                                break;
                        case 2:l.addatbeg();
                                break;
```

```
                case 3:l.addafter();
                        break;
                case 4:l.del();
                        break;
                case 5:l.display();
                        break;
                case 6:exit(0);
                        break;
                default:cout<<"invalid choice \n";
                        break;
            }
        }
        while(ch!=6);
        getch();
}
```

Output:

append:



Display:

```
enter your choice
1
enter the number
3

 MENU
1.append
2.add at begining
3.add after
4.delete
5.display
6.exit
enter your choice
5

1 2 3
 MENU
1.append
2.add at begining
3.add after
4.delete
5.display
6.exit
enter your choice
```

Add at beginning:

```
enter your choice
2
enter the number
9

 MENU
1.append
2.add at begining
3.add after
4.delete
5.display
6.exit
enter your choice
5

9 1 2 3
 MENU
1.append
2.add at begining
3.add after
4.delete
5.display
6.exit
enter your choice
_
```

Add after:

```
3
enter thelocation and no
2
8

 MENU
1.append
2.add at begining
3.add after
4.delete
5.display
6.exit
enter your choice
5

9 1 8 2 3
 MENU
1.append
2.add at begining
3.add after
4.delete
5.display
6.exit
enter your choice
_
```

Delete:

```
4

 enter the number
1

 MENU
1.append
2.add at begining
3.add after
4.delete
5.display
6.exit
enter your choice
5

9 8 2 3
 MENU
1.append
2.add at begining
3.add after
4.delete
5.display
6.exit
enter your choice
_
```

11.Create a binary search tree out of given data and traverse it inorder

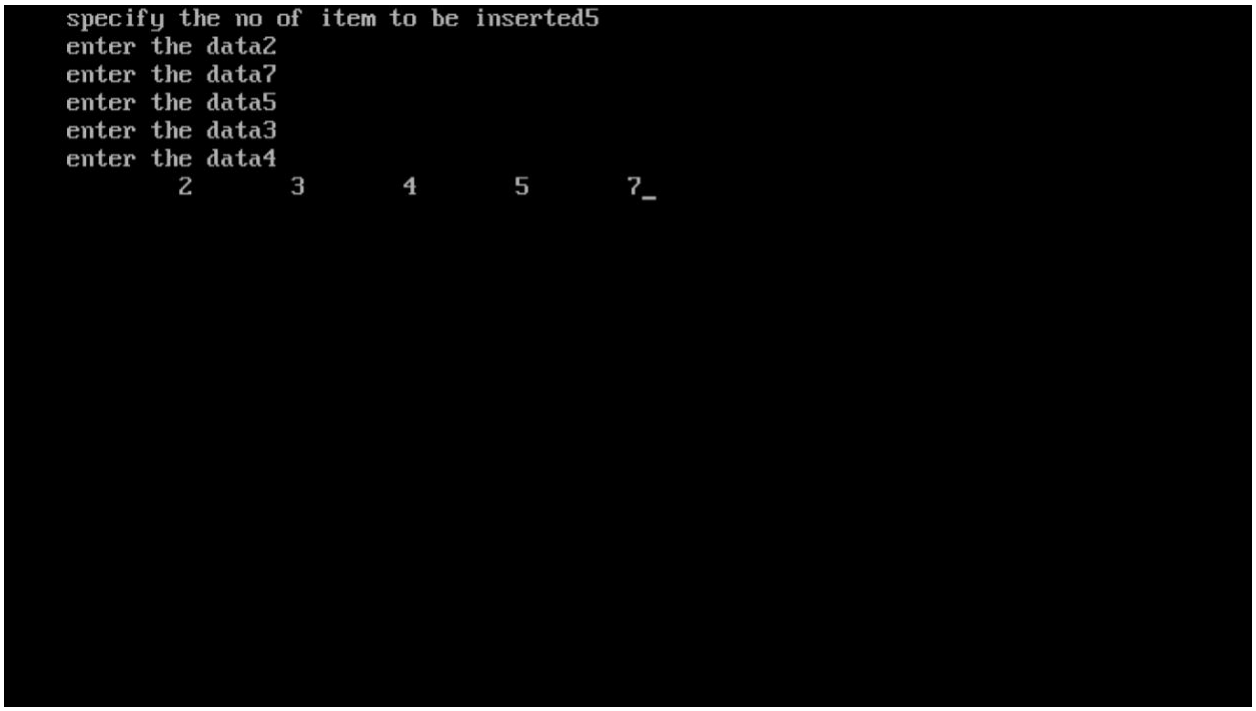
Program

```
#include<iostream.h>

#include<conio.h>

class btree
{
        public:struct btreenode

        {

                btreenode*leftchild;

                int data;

                btreenode*rightchild;

        }*root;

        public:

                btree();

                void buildtree(int num);

                static void insert(btreenode**sr,int num);

                void traverse();

                static void inorder(btreenode*sr);

};

btree::btree()

{

        root=NULL;

}

void btree::buildtree(int num)

{

        insert(&root,num);
```

```
}
void btree::insert(btreenode**sr,int num)
{
        if(*sr==NULL)
        {
                *sr=new btreenode;
                (*sr)->leftchild=NULL;
                (*sr)->data=num;
                (*sr)->rightchild=NULL;
                return;
        }
        else
        {
                if(num<(*sr)->data)
                insert(&((*sr)->leftchild),num);
                else
                insert(&((*sr)->rightchild),num);
        }
        return;
}
void btree::inorder(btreenode*sr)
{
        if(sr!=NULL)
        {
                inorder(sr->leftchild);
```

```
                cout<<"\t"<<sr->data;
                inorder(sr->rightchild);
        }
        else
                return;
}
void main()
{
        clrscr();
        btree bt;
        int req,i=1,num;
        cout<<"specify the num of item to be inserted\n";
        cin>>req;
        while(i++<=req)
        {
                cout<<"enter the data\t";
                cin>>num;
                bt.buildtree(num);
        }
        bt.inorder(bt.root);
        getch();
}
```

Output:


specify the no of item to be inserted5
enter the data2
enter the data7
enter the data5
enter the data3
enter the data4
2 3 4 5 7_

12. Program to merge two linked list

Program

```
#include<iostream.h>
#include<conio.h>
class linklist
{
        private:struct node
        {
                int data;
                node *link;
        }
        *p;
        public:linklist();
        void append(int);
        void merge(linklist&l,linklist&);
        void display();
        int count();
        ~linklist();
};
linklist::linklist()
{
        p=NULL;
}
void linklist::append(int num)
{
        node *temp;
```

```
        temp=p;
        if(temp==NULL)
        {
                temp=new node;
                p=temp;
        }
        else
        {
                while(temp->link!=NULL)
                temp=temp->link;
                temp->link=new node;
                temp=temp->link;
        }
        temp->data=num;
        temp->link=NULL;
}
void linklist::merge(linklist&l1,linklist&l2)
{
        while(l1.p!=NULL&&l2.p!=NULL)
        {
                if(l1.p->data<l2.p->data)
                {
                        append(l1.p->data);
                        l1.p=l1.p->link;
                }
                else if(l1.p->data>l2.p->data)
```

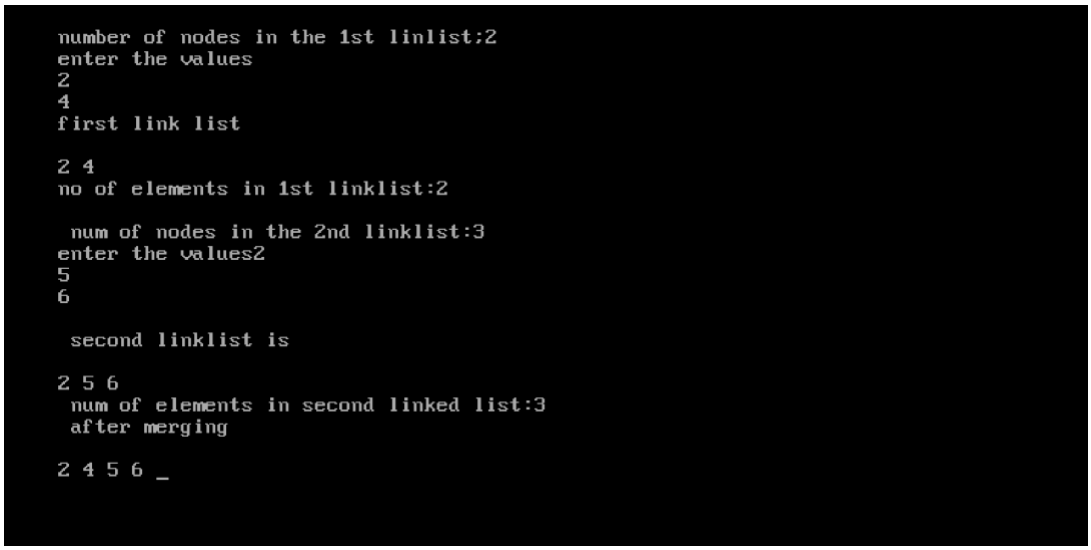
```
        {
                append(l2.p->data);
                l2.p=l2.p->link;
        }
        else
        {
                append(l1.p->data);
                l1.p=l1.p->link;
                l2.p=l2.p->link;
        }
}
if(l1.p!=NULL)
{
        while(l1.p!=NULL)
        {
                append(l1.p->data);
                l1.p=l1.p->link;
        }
}
else
{
        while(l2.p!=NULL)
        {
                append(l2.p->data);
                l2.p=l2.p->link;
        }
```

```
        }
}
void linklist::display()
{
        cout<<endl;
        node *temp=p;
        while(temp!=NULL)
        {
                cout<<temp->data<<" ";
                temp=temp->link;
        }
}
int linklist::count()
{
        int c=0;
        node *temp=p;
        while(temp!=NULL)
        {
                temp=temp->link;
                c++;
        }
        return c;
}
linklist::~linklist()
{
        node *q;
```

```
        while(p!=NULL)
        {
                q=p->link;
                delete p;
                p=q;
        }
}
void main()
{
        linklist l;
        int num,i,ch,n1,n2;
        clrscr();
        cout<<"\n number of nodes in the 1st linked list;";
        cin>>n1;
        cout<<"enter the values"<<endl;
        for(i=0;i<n1;i++)
        {
                cin>>num;
                l.append(num);
        }
        cout<<"first linked list"<<endl;
        l.display();
        cout<<"\nno of elements in 1st linked list:"<<l.count()<<"\n";
        linklist l2;
        cout<<"\n num of nodes in the 2nd linked list:";
        cin>>n2;
```

```
        cout<<"enter the values";
        for(i=0;i<n2;i++)
        {
                cin>>num;
                l2.append(num);
        }
        cout<<"\n second linked list is\n";
        l2.display();
        cout<<"\n num of elements in second linked list:"<<l2.count();
        linklist l3;
        l3.merge(l,l2);
        cout<<"\n after merging\n";
        l3.display();
        getch();
}
```

Output:



13. Program to add and delete elements from a queue

Program

```
#include<iostream.h>
#include<conio.h>
#include<process.h>
const int max=20;
class queue
{
        int rear,front,arr[max],item;
        public:queue()
        {
                rear=-1;
                front=-1;
        }
        void addq();
        void delq();
        void display();
};
void queue::addq()
{
        int t;
        if(rear==max-1)
        cout<<"queue is full\n";
        else
        {
                rear=rear+1;
```

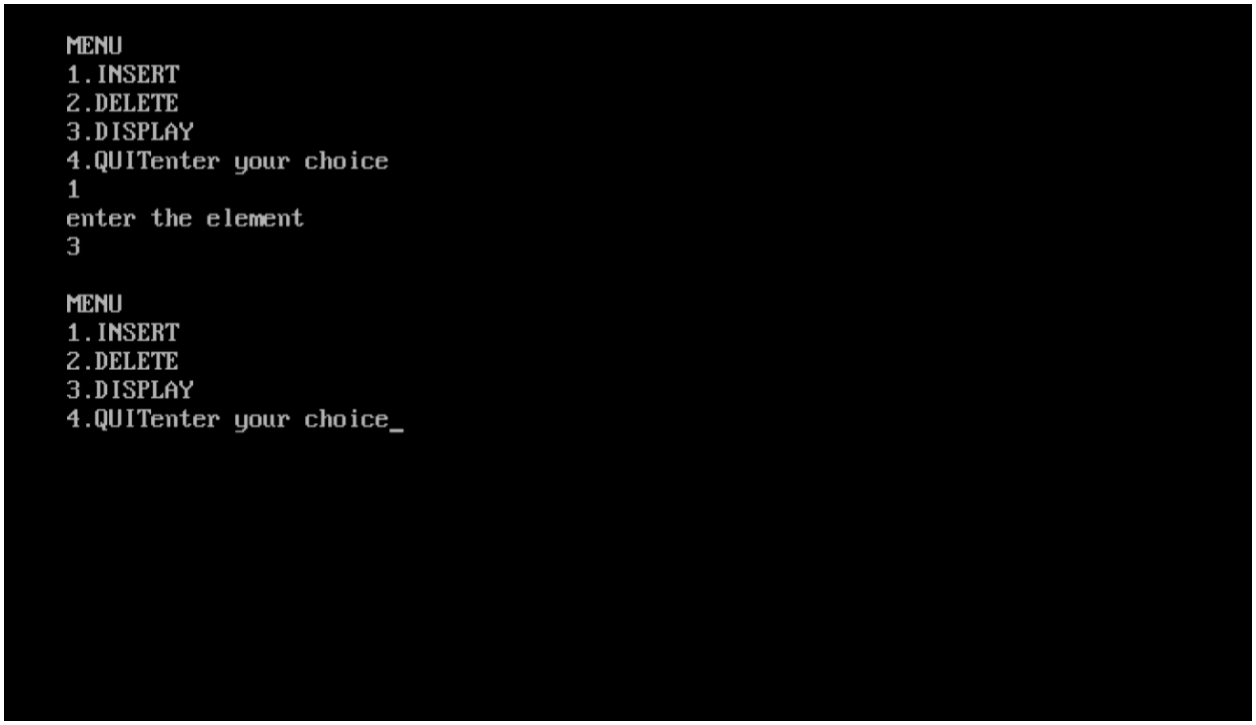
```
                cout<<"enter the elements\n";
                cin>>item;
                arr[rear]=item;
                if(front==-1)
                front++;
        }
}
void queue::delq()
{
        int data;
        if(rear==-1)
        {
                cout<<"queue is empty\n";
        }
        else
        {
                data=arr[front];
                arr[front]=0;
                if(front==rear)
                front=rear=-1;
                else
                front=front+1;
                cout<<"\n the deleted element is"<<data;
        }
}
void queue::display()
```

```
{
        for(int i=front;i<=rear;i++)
        {
                cout<<arr[i]<<"\t";
        }
}
int main()
{
        int n,i,data,ch;
        queue q;
        clrscr();
        do
        {
                cout<<"MENU\n1.INSERT\n2.DELETE\n3.DISPLAY\n4.QUIT";
                cout<<"\nenter your choice\n";
                cin>>ch;
                switch(ch)
                {
                        case 1:q.addq();
                        break;
                        case 2:q.delq();
                        break;
                        case 3:q.display();
                        break;
                        case 4:exit(0);
                        default:cout<<"invalid operator\n";
```

```
                    break;
                }
        }
        while(ch!=4);
        getch();
        return 0;
}
```

Output:

insert:



Display:

```
MENU
1.INSERT
2.DELETE
3.DISPLAY
4.QUITenter your choice1
enter the element
5

MENU
1.INSERT
2.DELETE
3.DISPLAY
4.QUITenter your choice
3
5
MENU
1.INSERT
2.DELETE
3.DISPLAY
4.QUITenter your choice
```

Delete:

```
MENU
1.INSERT
2.DELETE
3.DISPLAY
4.QUITenter your choice1
enter the element
4

MENU
1.INSERT
2.DELETE
3.DISPLAY
4.QUITenter your choice2
,the deleted element is4
MENU
1.INSERT
2.DELETE
3.DISPLAY
4.QUITenter your choice
```

14 .Operations in circular linked list

Program

```
#include<iostream.h>
#include<conio.h>
#include<process.h>
class linklist
{
        struct node
        {
                int data;
                node *link;
        }*p,*last;
        public:
                linklist();
                void append();
                void addatbeg();
                void addafter();
                void display();
                void del();
                ~linklist();
};
linklist::linklist()
{
        p=NULL;
        last=NULL;
}
```

```
void linklist::append()
{
        node *temp;
        int n;
        cout<<"enter the number \n";
        cin>>n;
        temp=new node;
        if(p==NULL)
        {
                p=temp;
                temp->data=n;
                temp->link=temp;
                last=temp;
        }
        else
        {
                temp->data=n;
                temp->link=last->link;
                last->link=temp;
                last=temp;
        }
}
void linklist::addatbeg()
{
        int n;
        node *temp;
```

```
        cout<<"enter the number \n";
        cin>>n;
        temp=new node;
        temp->data=n;
        temp->link=p;
        last->link=temp;
        p=temp;
}
void linklist::addafter()
{
        int n,loc;
        node *temp,*r;
        temp=p;
        cout<<"enter the location and number \n";
        cin>>loc>>n;
        for(int i=0;i<loc-1;i++)
        {
                temp=temp->link;
                if(temp==NULL)
                {
                        cout<<"there are less than"<<loc<<"elements list\n";
                }
        }
        r=new node;
        r->data=n;
        r->link=temp->link;
```

```
        temp->link=r;
}
void linklist::display()
{
        node *temp=p;
        cout<<endl;
        while(temp->link!=p)

        {
                cout<<temp->data<<" ";
                temp=temp->link;
        }
        cout<<temp->data<<"  ";
}
void linklist::del()
{
        int n;
        node *old,*temp;
        temp=p;
        cout<<"\nenter the number to be deleted\n";
        cin>>n;
        while(temp!=NULL)
        {
                if(temp->data==n)
                {
                        if(temp==p)
```

```
                {
                        p=temp->link;
                        last->link=temp;
                }
                else
                {
                old->link=temp->link;
                if(last==temp)
                        old->link=p;
                delete temp;
                return;
                }
        }
        else
        {
                old=temp;
                temp=temp->link;
        }
    }
    cout<<"element"<<n<<"not found\n";
}
linklist::~linklist()
{
    node *q;
    while(p!=NULL)
    {
```

```
                q=p->link;
                delete p;
                p=q;
        }
}
void main()
{
        linklist l;
        int ch;
        clrscr();
        do
        {
                cout<<"\nMENU\n";
                cout<<"1.Append\n";
                cout<<"2.Add at beginning\n";
                cout<<"3.Add after\n";
                cout<<"4.Delete \n";
                cout<<"5. Display\n";
                cout<<"6.Exit\n";
                cout<<"enter your choice\n";
                cin>>ch;
                switch(ch)
                {
                        case 1:l.append();
                                break;
                        case 2:l.addatbeg();
```

```
                    break;
                case 3:l.addafter();
                    break;
                case 4:l.del();
                    break;
                case 5:l.display();
                    break;
                case 6:exit(0);
                    break;
                default:cout<<"invalid choice\n";
                    break;
            }
        }
        while(ch!=6);
        getch();
}
```

Output:

append:

Display:

```
enter your choice
1
enter the number
5

MENU
1.Append
2.Add at beginning
3.Add after
4.Delete
5. Display
6.Exit
enter your choice
5

2 3 5
MENU
1.Append
2.Add at beginning
3.Add after
4.Delete
5. Display
6.Exit
enter your choice
```

Add at beginning:

```
enter your choice
2
enter the number
6

MENU
1.Append
2.Add at beginning
3.Add after
4.Delete
5. Display
6.Exit
enter your choice
5

6 2 3 5
MENU
1.Append
2.Add at beginning
3.Add after
4.Delete
5. Display
6.Exit
enter your choice
```

Add after :

```
3
enter the location and number
2
6

MENU
1.Append
2.Add at beginning
3.Add after
4.Delete
5. Display
6.Exit
enter your choice
5

6 2 6 3 5
MENU
1.Append
2.Add at beginning
3.Add after
4.Delete
5. Display
6.Exit
enter your choice
```

Delete:

```
4

enter the number to be deleted
2

MENU
1.Append
2.Add at beginning
3.Add after
4.Delete
5. Display
6.Exit
enter your choice
5

6 6 3 5
MENU
1.Append
2.Add at beginning
3.Add after
4.Delete
5. Display
6.Exit
enter your choice
```

15.Use a linked stack to reverse a string

Program

```
#include<iostream.h>
#include<conio.h>
#include<process.h>
#include<stdio.h>
class stack
{
        struct node
        {
                char DATA;
                struct node *link;
        }
        *top;
        public:stack();
                void push();
                void reverse();
};
stack::stack()
{
        top=NULL;
}
void stack::push()
{
        char str[50];
```

```
        cout<<"enter a string \n";
        gets(str);
        for(int i=0;str[i]!='\0';i++)
        {
                node *temp;
                temp=new node;
                if(temp==NULL)
                {
                        cout<<"overflow";
                        return;
                }
                temp->DATA=str[i];
                temp->link=top;
                top=temp;
        }
}
void stack::reverse()
{
        if(top==NULL)
        {
                cout<<"underflow";
                return;
        }
        char rev[50];
        for(int i=0;top!=NULL;i++)
```

```
        {
                rev[i]=top->DATA;
                top=top->link;
        }
        puts(rev);
}
void main()
{
        stack ob;
        clrscr();
        ob.push();
        ob.reverse();
        getch();
}
```

Output:

SQL-1

Create a table students with field sno,sname,sex,mark with sno as primary key and assign suitable constraints for each attribute. Insert five records into the table.

```
mysql> create table students(sno int,sname char(10),sex char(10),mark int,primary key(sno));
mysql> create table students(sno int,sname char(10),sex char(10),mark int,primary key(sno));
mysql> insert into students values(101,'anu','female',28);
mysql> insert into students values(103,'amal','male',34);
mysql> insert into students values(102,'suni','male',30);
mysql> insert into students values(104,'amala','female',44);
mysql> insert into students values(105,'arjun','male',44);
mysql> select *from students;
```

OUTPUT

```
+-----+--------+---------+-------+
| sno  | sname | sex      |mark |
+-----+--------+---------+------+
| 101 | anu     | female |   28   |
| 102 | suni    | male    |   30   |
| 103 | amal   | male    |   34   |
| 104 | amala | female |   44   |
| 105 | arjun  | male    |   44   |
+-----+--------+--------+-------+
```

1.Alter table by adding one more field rank.

```
mysql> alter table students add grade char(20);
mysql> select *from students;
```

OUTPUT

```
+-----+-------+--------+------+----------+
| sno | sname | sex    | mark | grade    |
+-----+-------+--------+------+----------+
| 101 | anu     | female |   28 | NULL  |
| 102 | suni    | male    |   30 | NULL  |
| 103 | amal   | male    |   34 | NULL  |
| 104 | amala | female |   44 | NULL  |
| 105 | arjun  | male    |   44 | NULL   |
+-----+-------+---------+-----+----------+
```

2.Find average mark.

```
mysql> select avg(mark) from students;
```

OUTPUT

```
+-------------+
| avg(mark) |
+-------------+
|   36.0000  |
+-------------+
```

3.Display sno and sname for those who have marks greater than average mark.

```
mysql> select sno,sname from students where mark>(select avg(mark) from students);
```

OUTPUT

```
+-----+--------+
| sno | sname |
+-----+--------+
| 104 | amala |
| 105 | arjun  |
+-----+--------+
```

4.Display the sname and sex whose sno is 104.

```
mysql> select sno,sname from students where sno=104;
```

OUTPUT

```
+-----+-------+
| sno | sname |
+-----+-------+
| 104 | amala |
+-----+-------+
```

5.Display all boys students with their name.

```
mysql> select sname,sex from students where sex='male';
```

OUTPUT

```
+-------+-------+
| sname | sex  |
+-------+-------+
| suni   | male |
| amal  | male |
| arjun | male |
+-------+-------+
```

SQL-2

Create a table department with field ename, salary, dno,dname,place with dno as primary key.

```
mysql> create table dept(dno int,ename char(10),salary int,dname char(30),place char(20),primary key(dno));

mysql> insert into dept values(1,'nikhil',10000,'b-unit','puranjan');
mysql> insert into dept values(2,'rajan',20000,'b-unit','arabikulam');
mysql> insert into dept values(3,'albin',30000,'b-unit','vilamana');
mysql> insert into dept values(4,'akhil',40000,'b-unit','puranjan');
mysql> insert into dept values(5,'josemon',50000,'b-unit','pulikurumba');
mysql> select *from dept;
```

OUTPUT

```
+-----+---------+--------+--------+----------------+
| dno | ename   | salary | dname  | place          |
+-----+---------+--------+--------+----------------+
|   1 | nikhil  |  10000 | b-unit | puranjan       |
|   2 | rajan   |  20000 | b-unit | arabikulam     |
|   3 | albin   |  30000 | b-unit | vilamana       |
|   4 | akhil   |  40000 | b-unit | puranjan       |
|   5 | josemon |  50000 | b-unit | pulikurumba    |
+---+-----------+---------+--------+---------------+
```

1.Rename the field with city.

```
mysql> select place as city from dept;
```

OUTPUT

```
+----------------+
| city           |
+----------------+
| puranjan       |
| arabikulam     |
| vilamana       |
| puranjan       |
| pulikurumba    |
+----------------+
```

2.Display the employee who got salary more than 6000 and less than 30000.

```
mysql> select ename from dept where salary>6000 and salary<30000;
```

OUTPUT

```
+----------+
| ename    |
+----------+
| nikhil   |
| rajan    |
+----------+
```

3.Display the total salary of the organization.

```
mysql> select sum(salary) from dept;
```

OUTPUT

```
+---------------+
| sum(salary) |
+---------------+
|      150000   |
+---------------+
```

4.Display ename for those who getting salary as between 5000 and 30000.

```
mysql> select ename from dept where salary>5000 and salary<30000;
```

OUTPUT

```
+---------+
| ename  |
+---------+
| nikhil   |
| rajan    |
+---------+
```

SQL-3

Create a table emp with fields eno,ename,job,dno,salary with eno as primary key. Insert five records into the table.

```
mysql> create table emp(eno int, ename char(20),job char(20),dno int,salary int,primary key(eno));
mysql> insert into emp values(10,'anu','manager',201,20000);
mysql> insert into emp values(11,'anusree','stock',202,20000);
mysql> insert into emp values(12,'sunil','stock',203,18000);
mysql> insert into emp values(13,'raju','accountant',204,18000);
mysql> insert into emp values(14,'rani','accountant',205,18000);
mysql> select * from emp;
```

OUTPUT

```
+-----+---------+-------------+------+----------+
| eno | ename   | job         | dno  | salary   |
+-----+---------+-------------+------+----------+
|  10 | anu     | manager     |  201 |  20000   |
|  11 | anusree | stock       |  202 |  20000   |
|  12 | sunil   | stock       |  203 |  18000   |
|  13 | raju    | accountant  |  204 |  18000   |
|  14 | rani    | accountant  |  205 |  18000   |
+-----+---------+-------------+------+----------+
```

1.Display ename,salary . salary with ascending order?

```
mysql> select ename,salary from emp order by salary asc;
```

OUTPUT

```
+---------+-----------+
| ename   | salary    |
+---------+-----------+
| sunil   |  18000    |
| raju    |  18000    |
| rani    |  18000    |
| anu     |  20000    |
| anusree |  20000    |
+---------+-----------+
```

2.Display ename and salary for eno=13?

```
mysql> select ename,salary from emp where eno=13;
```

OUTPUT

```
+-------+--------+
| ename | salary |
+-------+--------+
| raju  |  18000 |
+-------+--------+
```

3.Create another table department with field dno,dname,dmanager,place,with eno as primary key?

```
mysql> create table department(dno int,dname char(20),dmanager char(20),place char(20),primary key(dno));
mysql> insert into department values(101,'accountant','sunil','ckpara');
mysql> insert into department values(102,'clerk','sandra','ckpara');
mysql> insert into department values(103,'manager','sanoop','skpm');
mysql> insert into department values(104,'manager','don','skpm');
mysql> insert into department values(105,'store','dona','knr');
mysql> select * from department;
```

OUTPUT

```
+-----+------------+----------+----------+
| dno | dname      | dmanager | place  |
+-----+------------+----------+----------+
| 101 | accountant | sunil      | ckpara |
| 102 | clerk          | sandra     | ckpara |
| 103 | manager    | sanoop   | skpm   |
| 104 | manager    | don        | skpm   |
| 105 | store          | dona       | knr      |
+-----+------------+----------+----------+
```

4.Display the dmanager for the accountant department?

```
mysql> select dmanager from department where dname='accountant';
```

OUTPUT

```
+-------------+
| dmanager |
+----------+
| sunil        |
+------------+
```

SQL-4

Create a table emp with field with field eno,ename,job,manager,salary,with eno as primary key, Insert five records into the field.

```
mysql> create table emp(eno int,ename char(20),job char(20),manager char(20),salary int,primary
key(eno));
mysql> insert into emp values(101,'thomas','clerk','sunil',18000);
mysql> insert into emp values(102,'sree','accountant','sunil',19000);
mysql> insert into emp values(103,'vidya','peon','manu',20000);
mysql> insert into emp values(104,'varun','software','manu',20000);
mysql> insert into emp values(105,'varda','hardware','sanu',20000);
mysql> select * from emp;
```

OUTPUT

```
+------+----------+---------------+-------------+----------+
| eno | ename  | job           | manager  | salary   |
+------+----------+------------- --+-------------+----------+
| 101 | thomas | clerk          | sunil       |  18000   |
| 102 | sree      | accountant | sunil       |  19000   |
| 103 | vidya    | peon          | manu      |  20000   |
| 104 | varun   | software     | manu     |  20000    |
| 105 | varda   | hardware    | sanu      |  20000   |
+------+----------+----------------+-----------+-------------+
```

1.Display ename,salary from emp who are getting salary more than average salary of the organization.

```
mysql> select ename,salary from emp where salary>(select avg(salary)from emp);
```

OUTPUT

```
+--------+---------+
| ename | salary  |
+--------+---------+
| vidya  |  20000 |
| varun  |  20000 |
| varda  |  20000 |
+--------+---------+
```

2.Add 20%DA as extra salary to all employees label the column as "New Salary".

```
mysql> alter table emp add newsalary  int;
mysql> update emp set newsalary=salary+((salary*20)/100);
```

```
mysql> select * from  emp;
```

OUTPUT

```
+-----+----------+--------------+---------+---------+------------+
| eno | ename    | job          | manager| salary  | newsalary |
+-----+----------+--------------+---------+---------+-----------+
| 101 | thomas   | clerk        | sunil     |  18000 |     21600   |
| 102 | sree     | accountant   | sunil     |  19000 |     22800   |
| 103 | vidya    | peon         | manu    |  20000 |     24000   |
| 104 | varun    | software     | manu    |  20000 |     24000   |
| 105 | varda    | hardware     | sanu    |  20000 |     24000   |
+-----+----------+--------------+---------+---------+-----------+
```

3.Display the eno,ename,for all employees who got more than the average salary.Sort in descending order of salary?

```
mysql> select  ename ,salary from  emp  where  salary<(select avg(salary) from  emp)order  by  salary desc;
```

OUTPUT

```
+---------+---------+
| ename  | salary  |
+---------+---------+
| sree     |  19000 |
| thomas |  18000 |
+---------+---------+
```

4.Display ename,salary and salary with ascending order?

```
mysql> select ename,salary from emp order by salary asc;
```

OUTPUT

```
+--------+----------+
| ename | salary   |
+--------+----------+
| thomas |  18000 |
| sree     |  19000 |
| vidya    |  20000 |
| varun   |  20000  |
| varda   |  20000  |
+--------+----------+
```

5.Display ename job for manager=manu

```
mysql> select ename ,job from emp where manager='manu';
```

OUTPUT

```
+--------+------------+
| ename | job         |
+--------+------------+
| vidya  | peon        |
| varun  | software    |
+-------+-------------+
```

6.Display eno,ename from emp who work in a department with any employee who name contain the letter T.

```
mysql> select eno,ename from  emp  where  ename  like'%t%';
```

OUTPUT

```
+-----+--------+
| eno | ename  |
+-----+--------+
| 101 | thomas |
+-----+--------+
```

Sql-5

Create a table department with field dname,dno,salary,designation,ename,place with dno as primary key. Insert five records into the table.

```
mysql> create table department(dno int,ename char(10),salary int,designation char(20),dname
char(20),place char(20),primary key(dno));
mysql> inser t into department values(101,'anil',20000,'clerk','school','sfd');
mysql> insert  into department values(102,'anila',18000,'poen','school','wer');
mysql> insert  into department values(103,'arjun',80000,'manager','bank','xyz');
 mysql> insert into department values(104,'arun',80000,'accounting','bank','dfg');
mysql> insert  into department values(105,'aruna',28000,'teacher','school','dfg');
mysql> select *from department;
```

OUTPUT

```
+-----+--------+--------+---------------+--------+-------+
| dno | ename | salary | designation | dname  | place |
+-----+--------+--------+--------------+---------+-------+
| 101 | anil    |  20000 | clerk          | school | sfd     |
| 102 | anila  |  18000 | poen          | school | wer     |
| 103 | arjun  |  80000 | manager      | bank   | xyz    |
| 104 | arun   |  80000 | accounting  | bank   | dfg     |
| 105 | aruna |  28000 | teacher       | school | dfg     |
+-----+-------+--------+--------------+--------+---------+
```

1.Display the ename,dno for all employees who got more than average salary,sort the salary in descending order of salary?

```
mysql> select ename,dno from department where salary>(select avg(salary) from department) order by
salary desc;
```

OUTPUT

```
+--------+-----+
| ename | dno |
+--------+-----+
| arjun  | 103  |
| arun   | 104  |
+-------+------+
```

2.Display all employees who got salary between 15000 and 25000

```
mysql> select ename from department where salary>15000 and salary<25000;
```

OUTPUT

```
+--------+
| ename |
+--------+
| anil     |
| anila   |
+--------+
```

SQL-6

Create table students with fields sno,sname,mark with sno as primary key and assign suitable constraints for each attribute. Insert five records into the table.

```
mysql> create table students(sno int,sname char(10),sex char(10),mark int,primary key(sno));
mysql> insert  into  students  values(1,'amal','male',45);
mysql> insert  into  students  values(2,'akshara','female',41);
mysql> insert  into  students  values(3,'arun','male',42);
mysql> insert  into  students  values(4,'sandra','female',19);
mysql> insert  into  students  values(5,'akhila','female',40);
mysql> select * from  students;
```

OUTPUT

```
+-----+---------+--------+------+
| sno | sname   | sex    | mark |
+-----+---------+--------+------+
|   1 | amal    | male   |   45 |
|   2 | akshara | female |   41 |
|   3 | arun    | male   |   42 |
|   4 | sandra  | female |   19 |
|   5 | akhila  | female |   40 |
+---+----------+--------+------+
```

1.Alter the table by adding one more field rank.

```
mysql> alter  table s tudents  add  rank_ char;
mysql> update  students  set  rank_='a' where  mark>=45;
mysql> update  students  set  rank_='b' where  mark>=40 and  mark<45;
mysql> update  students  set  rank_='c' where  mark>=10 and  mark<40;
mysql> select * from  students;
```

OUTPUT

```
+-----+---------+--------+------+-------+
| sno | sname   | sex    | mark | rank_ |
+-----+---------+--------+------+-------+
|   1 | amal    | male   |   45 | a     |
|   2 | akshara | female |   41 | b     |
|   3 | arun    | male   |   42 | b     |
|   4 | Sandra  | female |   19 | c     |
|   5 | akhila  | female |   40 | b     |
+---+-----------+---------+-----+-------+
```

2.Display all male students with their name.

```
mysql> select sname from students where sex='male';
```

OUTPUT

```
+--------+
| sname |
+--------+
| amal   |
| arun    |
+--------+
```

3.Find the average mark?

```
mysql> select avg(mark) from students;
```

OUTPUT

```
+-------------+
| avg(mark) |
+-----------+
|   37.4000  |
+-------------+
```

4.Create a query to display the sno and sname for all students who got more than the average mark. Sort the result in ascending order of marks?

```
mysql> select sno,sname,mark from students where mark>(select avg(mark) from students) order by
mark asc;
```

OUTPUT

```
+-----+---------+-------+
| sno | sname   | mark |
+-----+---------+-------+
|   5 | akhila     |   40   |
|   2 | akshara  |   41   |
|   3 | arun       |   42   |
|   1 | amal      |   45   |
+---+-----------+-------+
```

5.Display girl student name for those who have marks greater than 40 and less than 20.

```
mysql> select sname,mark from students where sex='female' and mark>40 or mark<20;
```

OUTPUT

```
+----------+-------+
| sname    | mark |
+----------+-------+
| akshara |   41   |
| sandra   |   19   |
+----------+-------+
```

SQL-7

Create table employee with field eno,ename,job,salary,dept_no . with eno as primary key. Insert five records into the table.

```
mysql> create table employee(eno int,ename char(20),job char(20),salary int,dept_no int,primary
key(eno));
mysql> insert  into employee  values(10,'sanu','teacher',21000,101);
mysql> insert  into employee  values(11,'albin','clerk',7000,102);
mysql> insert  into employee  values(12,'athira','poen',9000,103);
mysql> insert  into employee  values(13,'anu','teacher',19000,104);
mysql> insert  into employee  values(14,'anju','teacher',19000,105);
mysql> select * from employee;
```

OUTPUT

```
+-----+--------+---------+--------+----------+
| eno | ename  | job     | salary  | dept_no |
+----+---------+---------+---------+---------+
|  10 | sanu    | teacher |  21000 |     101  |
|  11 | albin   | clerk    |   7000  |     102  |
|  12 | athira | poen     |   9000  |     103  |
|  13 | anu     | teacher |  19000 |     104   |
|  14 | anju    | teacher |  19000 |     105   |
+-----+--------+---------+--------+----------+
```

1.Display the details of employee who got salary between 5000 and 10000.

```
mysql> select * from  employee  where  salary>5000  and  salary<10000;
```

OUTPUT

```
+-----+--------+-------+--------+---------+
| eno | ename  | job   | salary | dept_no |
+-----+--------+-------+--------+---------+
|  11 | albin   | clerk |   7000 |     102   |
|  12 | athira | poen |   9000  |    103    |
+-----+--------+-------+--------+---------+
```

2.Display the employee details with salary in descending order.

```
mysql> select * from employee order by salary desc;
```

OUTPUT

```
+-----+--------+---------+--------+----------+
| eno | ename  | job     | salary | dept_no |
+-----+--------+---------+--------+----------+
|  10 | sanu   | teacher |  21000 |     101  |
|  13 | anu    | teacher |  19000 |     104  |
|  14 | anju   | teacher |  19000 |     105  |
|  12 | athira | poen    |   9000 |     103  |
|  11 | albin  | clerk   |   7000 |     102  |
+-----+--------+---------+--------+----------+
```

3.Display the employee details whose dept_no is 102.

```
mysql> select * from employee where dept_no=102;
```

OUTPUT

```
+-----+-------+-------+--------+---------+
| eno | ename | job   | salary | dept_no |
+-----+-------+-------+--------+---------+
|  11 | albin | clerk |   7000 |     102 |
+-----+-------+-------+--------+---------+
```

4.Display eno,ename of employee where the first letter of the name is 'S'

```
mysql> select eno,ename from employee where ename like's%';
```

OUTPUT

```
+-----+-------+
| eno | ename |
+-----+-------+
|  10 | sanu  |
+-----+-------+
```

5.Display the total salary.

```
mysql> select sum(salary) from employee;
```

OUTPUT

```
+---------------+
| sum(salary) |
+---------------+
|       75000    |
+---------------+
```

SQL-8

Create table deposit with acc no,accname,place,bname,amount with accno as primary key. Insert five record into the table.

```
mysql> create table deposit(accno int,accname char(10),place char(20),bname char(20),amount int,
primary key(accno));
mysql> insert into deposit values(101,'anu','wer','abc',35000);
mysql> insert into deposit values(102,'edon','axc','abc',5000);
mysql> insert into deposit values(103,'teena','poi','rgnc',50600);
mysql> insert into deposit values(104,'tito','ert','mkl',8900);
mysql> insert into deposit values(105,'tomy','gtrt','mkl',9900);
mysql> select * from deposit;
```

OUTPUT

```
+-------+----------+---------+-------+---------+
| accno | accname | place | bname | amount |
+-------+---------+---------+-------+----------+
|   101  | anu      | wer     | abc    |  35000  |
|   102  | edon    | axc     | abc    |   5000   |
|   103  | teena   | poi     | rgnc   |  50600 |
|   104  | tito      | ert      | mkl    |   8900  |
|   105  | tomy   | gtrt     | mkl    |   9900  |
+-------+---------+--------+--------+---------+
```

1.Display bname with maximum amount.

```
mysql> select bname,max(amount)from deposit group by bname;
```

OUTPUT

```
+--------+-----------------+
| bname | max(amount) |
+--------+-----------------+
| abc     |      35000      |
| rgnc    |      50600      |
| mkl     |       9900      |
+-------+-----------------+
```

2.Add a new field bonus to deposite bonus to deposite table. Display the deposit details.

```
mysql> alter table deposit add bonus int;
mysql> select * from deposit;
```

OUTPUT

```
+-------+-----------+--------+--------+---------+----------+
| accno | accname | place  | bname | amount | bonus    |
+-------+-----------+-------+---------+---------+----------+
|  101  | anu       | wer   | abc     | 35000 |  NULL   |
|  102  | edon      | axc   | abc     |  5000  |  NULL  |
|  103  | teena     | poi   | rgnc    |  50600 |  NULL  |
|  104  | tito      | ert   | mkl     |  8900  |  NULL  |
|  105  | tomy      | gtrt  | mkl     |  9900  |  NULL  |
+-------+-----------+-------+---------+---------+----------+
```

3.Add bonus 200 to the new field. Display the details.

```
mysql> update deposit set bonus=200;
mysql> select * from deposit;
```

OUTPUT

```
+-------+-----------+-------+--------+---------+--------+
| accno | accname | place | bname | amount | bonus |
+-------+-----------+-------+--------+----------+-------+
|  101  | anu       | wer   | abc     | 35000 |  200  |
|  102  | edon      | axc   | abc     |  5000  |  200  |
|  103  | teena     | poi   | rgnc    |  50600 |  200  |
|  104  | tito      | ert   | mkl     |  8900  |  200  |
|  105  | tomy      | gtrt  | mkl     |  9900  |  200  |
+-------+-----------+-------+---------+---------+-------+
```

4.Display the deposit details whose name contain the letter ‘E’.

```
mysql> delete from deposit where accname like'%e%';
mysql> select * from deposit;
```

OUTPUT

```
+-------+-----------+--------+--------+---------+--------+
| accno | accname | place  | bname | amount | bonus  |
+-------+-----------+--------+--------+---------+--------+
|  101  | anu       | wer    | abc     | 35000 |  200   |
|  104  | tito      | ert    | mkl     |  8900  |  200   |
|  105  |  tomy     | gtrt   | mkl     |  9900  |  200   |
+-------+----------+--------+---------+----------+-------+
```

SQL-9

Create a table with attribute tid,tname,age,dept,salary,sex with primary key tid. Insert 5 records into the table.

```
mysql> create table teacher(tid int,tname char(10),age int,dept char(20),salary int,sex char(10),primary key(tid));
mysql> insert into teacher values(100,'anju',30,'commerce',20000,'female');
mysql> insert into teacher values(101,'anu',27,'science',25000,'female');
mysql> insert into teacher values(102,'sandeep',30,'science',30000,'male');
mysql> insert into teacher values(103,'gautham',32,'science',32000,'male');
mysql> insert into teacher values(104,'sarath',26,'commerce',27000,'male');
mysql> select * from teacher;
```

OUTPUT

```
+-----+---------+------+--------------+--------+---------+
| tid | tname   | age  | dept         | salary | sex     |
+-----+---------+------+--------------+--------+---------+
| 100 | anju    |  30  | commerce     |  20000 | female  |
| 101 | anu     |  27  | science      |  25000 | female  |
| 102 | sandeep |  30  | science      |  30000 | male    |
| 103 | gautham |  32  | science      |  32000 | male    |
| 104 | sarath  |  26  | commerce     |  27000 | male    |
+-----+---------+------+--------------+--------+---------+
```

1.Display the name teacher with salary.

```
mysql> select tname,salary from teacher;
```

OUTPUT

```
+-----------+---------+
| tname     | salary  |
+-----------+---------+
| anju      |  20000  |
| anu       |  25000  |
| sandeep   |  30000  |
| gautham   |  32000  |
| sarath    |  27000  |
+-----------+---------+
```

2.Display the name of all male teacher.

```
mysql> select tname from teacher where sex='male';
```

OUTPUT

```
+-----------+
| tname     |
+-----------+
| sandeep  |
| gautham |
| sarath    |
+----------+
```

3.Display the name of teacher whose age>30.

```
mysql> select tname from teacher where age>30;
```

OUTPUT

```
+-----------+
| tname     |
+-----------+
| gautham |
+----------+
```

4.Display the average salary.

```
mysql> select avg(salary) from teacher;
```

OUTPUT

```
+----------------+
| avg(salary)    |
+----------------+
|  26800.0000 |
+----------------+
```

SQL-10

Create a table customer with field cid,cname,dob,place.

```
mysql> create table customer(cid int,cname char(20),dob char(20),place char(20),primary key(cid));
mysql> insert into customer values(1,'abhi','4-1-1999','chemperi');
mysql> insert into customer values(2,'albin','14-12-1999','paissakari');
 mysql> insert into customer values(3,'hiran','9-6-1998','chemperi');
mysql> insert into customer values(4,'vishnu','2-5-2000','payyavoor');
mysql> insert into customer values(5,'joel','1-11-1999','chamathachal');
mysql> select * from customer;
```

OUTPUT

```
+-----+--------+--------------+----------------+
| cid | cname  | dob          | place          |
+---+---------+--------------+----------------+
|   1 | abhi   | 4-1-1999     | chemperi       |
|   2 | albin  | 14-12-1999 | paissakari     |
|   3 | hiran  | 9-6-1998     | chemperi       |
|   4 | vishnu | 2-5-2000     | payyavoor      |
|   5 | joel   | 1-11-1999    | chamathachal |
+---+---------+--------------+----------------+
```

Create table loan with loan_no,cid,bname assigning suitable constraints.

```
mysql> create table loan(loan_no int,cid int,bname char(30),primary key(loan_no));
mysql> insert into loan values(1,1,'chemperi');
mysql> insert into loan values(2,2,'paissakari');
mysql> insert into loan values(3,3,'chemperi');
mysql> insert into loan values(4,4,'payyavoor');
mysql> insert into loan values(5,5,'chamathachal');
mysql> select *from loan;
```

OUTPUT

```
+----------+-----+----------------+
| loan_no | cid  | bname          |
+---------+------+----------------+
|      1    |   1  | chemperi       |
|      2    |   2  | paissakari     |
|      3    |   3  | chemperi       |
|      4    |   4  | payyavoor      |
|      5    |   5  | chamathachal |
+--------+-------+----------------+
```

Create table depositor with field acc_no,cid,balance,bname. Assingning suitable constraints.

```
mysql> create table depositor(acc_no int,cid int,balance float(20),bname char(20),primary key(acc_no));


mysql> insert into depositor values(123,1,5000,'chemperi');
mysql> insert into depositor values(124,2,6000,'paissakari');
mysql> insert into depositor values(125,3,7000,'chemperi');
mysql> insert into depositor values(126,4,8000,'payyavoor');
mysql> insert into depositor values(127,5,6500,'chamathachal');
 mysql> select * from depositor;
```

OUTPUT

```
+--------+------+---------+---------------+
| acc_no | cid  | balance | bname         |
+--------+------+---------+---------------+
|   123  |   1  |   5000 | chemperi       |
|   124  |   2  |   6000 | paissakari     |
|   125  |   3  |   7000 | chemperi       |
|   126  |   4  |   8000 | payyavoor      |
|   127  |   5  |   6500 | chamathachal   |
+--------+------+---------+---------------+
```

1.Add one more field amount to loan table. Update each record. Display cname for cid=2.

```
mysql> alter table loan add amount float(20);
mysql> update loan set amount=6000 where loan_no=001;
mysql> update loan set amount=5000 where loan_no=002;
mysql> update loan set amount=6500 where loan_no=003;
mysql> update loan set amount=7000 where loan_no=004;
mysql> update loan set amount=8000 where loan_no=005;
mysql> select cname,cid from customer where cid=2;
```

OUTPUT

```
+-------+-----+
| cname | cid |
+-------+-----+
| albin   |  2  |
+-------+-----+
```

2.Calculate Rs 150 extra for all customer having loan. The added loan amount will display in new coloumn.

```
mysql> alter table loan add new_amount float(20);
```

```
mysql> update loan set new_amount=loan.amount+150;
mysql> select * from loan;
```

OUTPUT

```
+----------+-----+----------------+---------+----------------+
| loan_no | cid  | bname          | amount | new_amount |
+----------+----+----------------+---------+----------------+
|      1     |   1 | chemperi       |  6000  |      6150      |
|      2     |   2 | paissakari     |  5000  |      5150      |
|      3     |   3 | chemperi       |  6500  |      6650      |
|      4     |   4 | payyavoor      |  7000  |      7150      |
|      5     |   5 | chamathachal |  8000  |      8150      |
+----------+----+----------------+---------+----------------+
```

3.Display loan_no,cname and place of a customer who is residing in chemperi.

```
mysql> select loan_no,cname,place from customer,loan where customer.cid=loan.cid and
place='chemperi';
```

OUTPUT

```
+---------+-------+-------------+
| loan_no | cname | place       |
+--------+---------+------------+
|      1    | abhi     | chemperi  |
|      3    | hiran   | chemperi   |
+--------+--------+-------------+
```

4.Display all information from loan table for loan_no 002,004,005.

```
mysql> select *from loan where loan_no in(002,004,005);
```

OUTPUT

```
+---------+------+----------------+----------+----------------+
| loan_no | cid  | bname          | amount | new_amount |
+---------+------+----------------+----------+----------------+
|      2     |   2   | paissakari      |  5000  |      5150      |
|      4     |   4   | payyavoor      |  7000  |      7150      |
|      5     |   5   | chamathachal |  8000  |      8150      |
+---------+------+----------------+---------+----------------+
```

4.Display all information from loan table for loan_no 002,004,005.

```
mysql> select cname from customer,loan,depositor where customer.cid=loan.cid=depositor.cid;
```

OUTPUT

```
+--------+
| cname  |
+--------+
| joel   |
| vishnu |
| hiran  |
| albin  |
| abhi   |
+--------+
```