LAB MANUAL
FOR
DATA STRUCTURE USING C LAB

PROGRAM NO.1

Aim: - To search an element in the array using Linear Search.

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int a[10],i,item,flag=0;
        clrscr();
        printf("Enter the data in the array");
        for(i=0;i<10;i++)
        {
                scanf("%d",&a[i]);
        }
        printf("Enter the element to be searched");
        scanf("%d",&item);
        for(i=0;i<10;i++)
        {
                if(item==a[i])
                {
                        flag=1;
                        break;
                }
        }
        if(flag==0)
        printf("Element Not Found");
        else
        printf("Element Found at Position =%d",i);
getch();
}
```

PROGRAM NO.2

Aim: - To search an element in the 2-dimensional array using Linear Search.

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int a[3][3],i,j,item,flag=0;
        clrscr();
        printf("Enter the data in the array");
        for(i=0;i<3;i++)
        {
        for(j=0;j<3;j++)
                {
                scanf("%d",&a[i][j]);
                }
        }
        printf("Enter the element to be searched");
        scanf("%d",&item);
        for(i=0;i<3;i++)
        {
        for(j=0;j<3;j++)
        {
                if(item==a[i][j])
                {
                        flag=1;
                        printf("Element found at position =%d,%d",i,j);
                }
        }
        }
        if(flag==0)
        printf("Element Not Found");

getch();
```

```
}
```

PROGRAM NO.3

Aim: - To merge two sorted array into one sorted array.

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int a[10],b[10],c[20],i,j,k,n,m,t;
        clrscr();
        printf("Enter size of Array A\n");
        scanf("%d",&n);
        printf("Enter the data in Array A\n");
        for(i=0;i<n;i++)
        {
        scanf("%d",&a[i]);
        }
        printf("Enter size of Array B\n");
        scanf("%d",&m);
        printf("Enter the data in Array B\n");
        for(j=0;j<m;j++)
        {
        scanf("%d",&b[j]);
        }
        i=j=k=0;
        while(i<n&&j<m)
        {
                if(a[i]<b[j])
                c[k++]=a[i++];
                else
                if(a[i]>=b[j])
                c[k++]=b[j++];
        }
```

```
        if(i<n)
        {
                for(t=0;t<n;t++)
                c[k++]=a[i++];
        }
        else
        {
                for(t=0;t<m;t++)
                  c[k++]=b[j++];
        }
          printf("\n");
          for(k=0;k<(m+n);k++)
          printf("\n %d ",c[k]);
  getch();
}
```

PROGRAM NO.4

Aim: - To perform the following operation in Matrix
1. Addition 2. Subtraction 3. Multiplication 4. Transpose

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int a[3][3],b[3][3],c[3][3],d[3][3],i,j,k;
        clrscr();
        printf("Enter the data in Matrix A");
        for(i=0;i<3;i++)
        {
                for(j=0;j<3;j++)
                {
                scanf("%d",&a[i][j]);
                }
        }
        printf("Enter the data in Martix B");
        for(i=0;i<3;i++)
        {
                for(j=0;j<3;j++)
                {
                scanf("%d",&b[i][j]);
                }
        }
        for(i=0;i<3;i++)
        {
                for(j=0;j<3;j++)
```

```
            {
            c[i][j]=a[i][j]+b[i][j];
            }
}
printf("Addition of two Matrix A and B is\n");
for(i=0;i<3;i++)
{
            for(j=0;j<3;j++)
            {
            printf("%d\t",c[i][j]);
            }
            printf("\n");
}
            for(i=0;i<3;i++)
{
            for(j=0;j<3;j++)
            {
            c[i][j]=a[i][j]-b[i][j];
            }
}
printf("Subtraction  of two Matrix A and B is\n");
for(i=0;i<3;i++)
{
            for(j=0;j<3;j++)
            {
            printf("%d\t",c[i][j]);
            }
            printf("\n");
}
printf("Transpose  of Matrix C is\n");
for(i=0;i<3;i++)
{
            for(j=0;j<3;j++)
            {
            d[j][i]=c[i][j];
            }
}
 for(i=0;i<3;i++)
{
```

```
            for(j=0;j<3;j++)
            {
            printf("%d\t",d[i][j]);
            }
            printf("\n");
      }
            printf("Multiplication of Matrix A and B is\n");

            for(i=0;i<3;i++)
            {
                  for(j=0;j<3;j++)
                  {
                  c[i][j]=0;
                  for(k=0;k<3;k++)
                  {
                   c[i][j]=c[i][j]+a[i][k]*b[k][j];
                  }
                  }
            }
            printf("\n");
            for(i=0;i<3;i++)
            {
            for(j=0;j<3;j++)
            {
            printf("%d\t",c[i][j]);
            }
            printf("\n");
            }
getch();
}
```

PROGRAM NO.5

Aim: - To perform the swapping of two numbers using call by value and call by reference.

```
#include<stdio.h>
#include<conio.h>
void swapbyvalue(int,int);
void swapbyref(int*,int*);
void main()
{
        int a,b;
        clrscr();
        printf("Enter the two numbers");
        scanf("%d%d",&a,&b);
        swapbyvalue(a,b);
        swapbyref(&a,&b);
        printf("\nNumber after swapping by Reference\n");
        printf("\na=%d\nb=%d",a,b);
        getch();
}
void swapbyvalue(int x, int y)
{
        int temp;
        temp=x;
        x=y;
```

```
        y=temp;
        printf("\nNumbers after swapping by value are\n");
        printf("a=%d",x);
        printf("\nb=%d",y);
}
void swapbyref(int *x,int *y)
{
        int temp;
        temp=*x;
        *x=*y;
        *y=temp;
}
```

PROGRAM NO.6

Aim: - To perform following operation on strings using string functions 1. Addition 2. Copying 3. Reverse 4. Length of String.

```
#include<conio.h>
#include<stdio.h>
#include<string.h>
void main()
{
        char a[20],b[20],c[20];
        int l;
        clrscr();
        printf("Enter the First String");
        scanf("%s",&a);
        printf("Enter the Second String");
        scanf("%s",&b);
        strcat(a,b);
        printf("\nConcatenation of String a and b is:%s",a);
        l=strlen(a);
        printf("\nLength of String is %d",l);
        strcpy(c,a);
        printf("\nthe Copied String is %s",c);
        strrev(a);
```

```
        printf("\nreverse of String is %s",a);

        getch();
}
```

PROGRAM NO.7 (a)

Aim: - To search an element in the array using Iterative Binary Search.

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int a[20],n,mid,beg,i,end,item,loc=-1;
        clrscr();
        printf("Enter the number of elements to be entered\n");
        scanf("%d",&n);
        printf("Enter the elements in ascending order");
        for(i=0;i<n;i++)
        {
                scanf("%d",&a[i]);
        }
        printf("Enter the element to be searched");
        scanf("%d",&item);
        beg=0;
        end=n-1;
        while(beg<=end)
```

```
        {
                mid=(beg+end)/2;
                if(item==a[mid])
                {
                loc=mid;
                break;
                }
        else if(a[mid]<item)
        beg=mid+1;
        else
        end=mid-1;
        }
        if(loc==-1)
        printf("Element Not Present");
        else
        printf("Element found at =%d",loc);
        getch();
 }
```

PROGRAM NO.7 (b)

Aim: - To search an element in the array using Recursive Binary Search.

```
#include<stdio.h>
#include<conio.h>
void binary(int [],int,int);
void main()
{
        int a[20],i,n,item;
        clrscr();
        printf("Enter the number of items in the array");
        scanf("%d",&n);
        printf("enter the data in array");
        for(i=0;i<n;i++)
        {
         scanf("%d",&a[i]);
        }
        printf("Enter the element to be searched");
```

```
        scanf("%d",&item);

        binary(a,n,item);
getch();
}

void binary(int a[],int n,int item)
{
 int beg,end,mid,loc=-1;
 beg=0;
 end=n-1;
 while(beg<=end)
   {
        mid=(beg+end)/2;
        if(item==a[mid])
        {
                loc=mid;
                break;
        }

else if(item>a[mid])
        beg=mid+1;
        else
        end=mid-1;
   }
   if(loc==-1)
   printf("Element not Found");
   else
   printf("Element Found at position = %d",loc);
}
```

PROGRAM NO.8

Aim: - To implement Bubble Sort.

```
#include<stdio.h>
#include<conio.h>
void bubble(int [],int);
void main()
{
        int a[20],i,n;
        clrscr();
        printf("Enter the number of items in the array");
        scanf("%d",&n);
        printf("Enter the data in the array");
        for(i=0;i<n;i++)
        {
         scanf("%d",&a[i]);
```

```
        }
        bubble(a,n);
getch();
}
void bubble(int a[],int n)
{
    int i,temp,j,p;
    for(i=1;i<n;i++)
    {
        for(p=0;p<n-i;p++)
           {
                if(a[p]>a[p+1])
                {
                        temp=a[p];
                        a[p]=a[p+1];
                        a[p+1]=temp;
                }
           }
    }
    for(i=0;i<n;i++)
    printf("\n%d",a[i]);
}
```

PROGRAM NO.9

Aim: - To implement Selection Sort.

```
#include<stdio.h>
#include<conio.h>
void select(int [],int);
void bubble(int [],int);
int min(int [],int,int);

void main()
{
        int a[20],i,n;
        clrscr();
        printf("Enter the number of items in the array");
```

```
        scanf("%d",&n);
        printf("Enter the data in the array");
        for(i=0;i<n;i++)
        {
                scanf("%d",&a[i]);
        }
        bubble(a,n);
        select(a,n);
        getch();
}
void bubble(int a[],int n)
{
    int i,temp,p;
    for(i=1;i<n;i++)
    {
        for(p=0;p<n-i;p++)
           {
                if(a[p]>a[p+1])
                {
                        temp=a[p];
                        a[p]=a[p+1];
                        a[p+1]=temp;
                }
           }
    }
    printf("\nData After Bubble Sort");
    for(i=0;i<n;i++)
    printf("\n%d",a[i]);
}

void select(int a[],int n)
{
        int i,loc,temp;
        loc=0;
        temp=0;
        for(i=0;i<n;i++)
        {
                loc=min(a,i,n);
                temp=a[loc];
```

```
                a[loc]=a[i];
                a[i]=temp;
        }
        printf("\nData After Selection Sort");
        for(i=0;i<n;i++)
        printf("\n%d",a[i]);
}

int min(int a[],int lb,int ub)
{
        int m=lb;
        while(lb<ub)
        {
                if(a[lb]<a[m])
                {
                m=lb;
                }
        lb++;
        }
        return m;
}
```

PROGRAM NO.10

Aim: - To implement Insertion Sort.

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

void insert(int [],int);
void main()
{
        int a[20],i,n;
```

```
        clrscr();
        printf("Enter the number of items in the array");
        scanf("%d",&n);
        printf("Enter the data in the array");
        for(i=0;i<n;i++)
        {
                scanf("%d",&a[i]);
        }
        insert(a,n);
        getch();
}


void insert(int a[],int n)
{
        int i,j,temp;
        for(i=1;i<n;i++)
        {
                temp=a[i];
                for(j=i-1;j>=0;j--)
                {
                        if(a[j]>temp)
                        {
                        a[j+1]=a[j];
                        }
                        else
                        break;
                }
                a[j+1]=temp;
        }
        printf("Data After Insertion Sort");
        for(i=0;i<n;i++)
        printf("\n%d",a[i]);

}
```

PROGRAM NO.11

Aim: - To implement Quick Sort.

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

void quicksort(int[],int,int);
int partition(int [],int,int);
```

```
void main()
{
        int a[20],i,n;
        clrscr();
        printf("Enter the size of array");
        scanf("%d",&n);
        printf("Enter the elements in the array");
        for(i=0;i<n;i++)
        {
                scanf("%d",&a[i]);
        }
        quicksort(a,0,n-1);
        for(i=0;i<n;i++)
        printf("\n%d",a[i]);
        getch();
}

void quicksort(int a[],int lb,int ub)
{
        int mid;
        if(lb<ub)
        {
                mid=partition(a,lb,ub);
                quicksort(a,lb,mid-1);
                quicksort(a,mid+1,ub);
        }
}


int partition(int a[],int lb,int ub)
{
        int i,p,q,t;
        p=lb+1;
        q=ub;
        i=a[lb];

        while(q>=p)
```

```
        {
                while(a[p]<i)
                p++;
                while(a[q]>i)
                q--;
                if(q>p)
                {
                        t=a[p];
                        a[p]=a[q];
                        a[q]=t;
                }
        }

        t=a[lb];
        a[lb]=a[q];
        a[q]=t;
        return q;
}
```


PROGRAM NO.12

Aim: - To implement Merge Sort.

```
#include<stdio.h>
#include<conio.h>
void mergesort(int a[],int,int);
void merge(int [],int,int,int);
void main()
{
        int a[20],i,n;
        clrscr();
        printf("Enter the number of elements");
        scanf("%d",&n);
        printf("Enter the elements");
        for(i=0;i<n;i++)
        {
        scanf("%d",&a[i]);
        }
        mergesort(a,0,n-1);
        printf("Data After Merge Sort");
        for(i=0;i<n;i++)
        printf("\n%d",a[i]);
        getch();
}
void mergesort(int a[],int lb,int ub)
{
        int mid;
        if(lb<ub)
        {
                mid=(lb+ub)/2;
                mergesort(a,lb,mid);
                mergesort(a,mid+1,ub);
                merge(a,lb,mid+1,ub);
        }
}


void merge(int a[],int lb,int mid,int ub)
{
        int k,p1,p2,p3,b[20];
        p1=lb;
        p3=lb;
```

```
        p2=mid;
        while((p1<mid)&&(p2<=ub))
        {
                if(a[p1]<=a[p2])

                b[p3++]=a[p1++];

                else

                b[p3++]=a[p2++];

        }
        while(p1<mid)
        {
                b[p3++]=a[p1++];
        }
        while(p2<=ub)
        {
                b[p3++]=a[p2++];
        }
        for(k=lb;k<p3;k++)
        {
        a[k]=b[k];
        }
}
```


PROGRAM NO.13

Aim: - To implement Stack using array.

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

void push();
void pop();
void display();

int top;
int a[5];

void main()
{
        int choice;
        char ch;
        top=-1;
        clrscr();
        do
        {
        printf("\n\t 1. PUSH");
        printf("\n\t 2. POP");
        printf("\n\t 3. DISPLAY");
        printf("\n\t 4. EXIT");
        printf("\nEnter your choice");
        scanf("%d",&choice);
        switch(choice)
        {
        case 1:
        push();
        break;
        case 2:
        pop();
        break;

        case 3:
        display();
```

```
        break;
        case 4:
        exit(0);
        default:
        printf("\nBAD CHOICE");
        }
        printf("\ndo you want to continue y/n");
        ch=getche();
        }
        while(ch=='y');
}

void push()
{
        int item;
        if(top==4)
        printf("STACK IS FULL");
        else
        {
                printf("Enter the item to be inserted");
                scanf("%d",&item);
                top=top+1;
                a[top]=item;
                //top=tope;
        }
}

void pop()
{
        int item;
        if(top==-1)
        printf("STACK IS EMPTY");
        else
        {
                item=a[top];
                top=top-1;
                printf("%d is deleted",item);
                //top=tope;
        }
```

```
}


void display()
{
        int i;
        for(i=top;i>=0;i--)
        printf("\n%d",a[i]);
}
```

PROGRAM NO.14

Aim: - To implement Queue using array.

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

void insert();
void delet();
void display();
int front,rear;
int q[5];

void main()
{
        int choice;
        char ch;
        front=-1;
        rear=-1;
        clrscr();
        do
        {
        printf("\n\t 1. INSERT");
        printf("\n\t 2. DELETE");
        printf("\n\t 3. DISPLAY");
        printf("\n\t 4. EXIT");
        printf("\nEnter your choice");
        scanf("%d",&choice);
        switch(choice)
        {
        case 1:
        insert();
        break;
        case 2:
        delet();
        break;
```

```
        case 3:
        display();
        break;
        case 4:
        exit(0);
        default:
        printf("\nBAD CHOICE");
        }
        printf("\ndo you want to continue y/n");
        ch=getche();
        }
        while(ch=='y'||'Y');
}

void insert()
{
        int item;
        if(((front==1)&&(rear==5))||(front==rear+1))
        {
        printf("QUEUE IS FULL");
        }
        else
        {
                printf("Enter the element");
                scanf("%d",&item);
                if(front==-1)
                {
                        front=1;
                        rear=1;
                }
                else if(rear==5)
                {
                        rear=0;
                }
                else
                {
                        rear=rear+1;
                }
                q[rear]=item;
```

```
        }
}

void delet()
{
        int item;
        if(front==-1)
        {
         printf("QUEUE IS EMPTY");
        }
        else
        {
                item=q[front];
                if(front==rear)
                {
                        front=-1;
                        rear=-1;
                }
                else if(front==5)
                {
                        front=0;
                }
                else
                front=front+1;
                printf("%d is deleted",item);
        }
}

void display()
{
        int i;
        if(front==-1)
        printf("QUEUE IS EMPTY");
        else
        {
        for(i=front;i<=rear;i++)
        {
        printf("\n%d",q[i]);
        }}
```

```
}
```

PROGRAM NO.15

Aim: - To implement Linked List.

```
#include<stdio.h>
#include<conio.h>
#include<alloc.h>
#include<process.h>

struct node
{
        int info;
        struct node *next;
};
struct node *start=NULL;

void ins();
void ins_at_beg
();
void ins_at_mid();
void ins_at_end();
void del();
void del_at_beg();
void del_at_mid();
void del_at_end();
void display();
int count();

void main()
{
        int ch=0,i=0,cnt;
        clrscr();
        while(1)
        {
                printf("***********menu***********");
                printf("\n1.insert");

                printf("\n2.delete");
```

```
                printf("\n3.display");
                printf("\n4.count");
                printf("\n5.exit");
                printf ("\nenter your choice : ");
                scanf("%d",&ch);


                switch(ch)
                {
                        case 1:ins();
                        break;
                        case 2:del();
                        break;
                        case 3:display();
                        break;
                        case 4:cnt=count();
                                printf("\n the no of nodes : %d\n",cnt);
                        break;
                        case 5:exit(1);

                }
        }
}

void ins()
{
        int j=0,ch1=0;
        printf("\nenter your choice");
        printf("\n1.insert at the beggning");
        printf("\n2.insert at the middle");
        printf("\n3.insert at the end");
        scanf ("%d",&ch1);
        switch(ch1)
        {
                case 1:ins_at_beg();
                break;
                case 2:ins_at_mid();
                break;
                case 3:ins_at_end();


        }
}
void ins_at_beg()
{
```

```
        int info;
        struct node *t=(struct node *)malloc(sizeof(struct node));
        printf("\nenter information to be inserted in the beggning");
        scanf("%d",&info);
        t->info=info;
        t->next=start;
        start=t;
}
void ins_at_mid()
{
        int inform,x,i;
        struct node *t=(struct node *)malloc(sizeof(struct node));
        struct node *p=start;
        printf("\nenter the location after which new node to be added");
        scanf("%d",&x);
        for(i=1;i<x;i++)
                p=p->next;
        printf("\nenter information of the new node");
        scanf("%d",&inform);
        t->info=inform;
        t->next=p->next;
        p->next=t;
}

void ins_at_end()
{
        int inform1;
        struct node *t=(struct node *)malloc(sizeof(struct node));
        struct node *p=start;

        printf("\nenter information to be added");
        scanf("%d",&inform1);
        t->info=inform1;
        while(p->next!=NULL)
                p=p->next;

        p->next=t;
        t->next=NULL;
}

void del()
{
        int k=0,ch2=0;
        printf("\nenter your choice");
        printf("\n1.delete at the beggning");
```

```
        printf("\n2.delete at the middle");
        printf("\n3.delete at the end");
        scanf ("%d",&ch2);
        switch(ch2)
        {
                case 1:del_at_beg();
                break;
                case 2:del_at_mid();
                break;
                case 3:del_at_end();
                break;
        }

}

void del_at_beg()
{
        struct node *t=start;
        start=start->next;
        free(t);
}

void del_at_mid()

{
        int n;
        struct node *cur=start;
        struct node *pre=start;
        printf("\nenter information to be deleted");
        scanf("%d",&n);
        while(cur->info!=n)
        {
                pre=cur;
                cur=cur->next;
        }
        pre->next=cur->next;
        free(cur);
}

void del_at_end()
{
        struct node *cur=start;
        struct node *pre=start;
        while(cur->next!=NULL)
        {
```

```
                pre=cur;
                cur=cur->next;
        }
        pre->next=NULL;
        free(cur);
}
void display()
{

        struct node *p=start;
        printf("\n\n**********LINK LIST**********\n\n");
        while(p!=NULL)
        {
                printf("%d\n",p->info);
                p=p->next;
        }


}
int count()
{
         int c=0;
         struct node *q=start;
         while(q!=NULL)
        {
                q=q->next;

                c=c+1;
        }
  return c;
}
```