



# DATA STRUCTURE

Sycse1

Nikhil  
chalikw  
ar

Digitally signed by  
Nikhil chalikwar  
DN: cn=Nikhil  
chalikwar, o,  
ou=sycse1,  
email=chalikwarnikhil  
@gmail.com, c=IN  
Date: 2018.11.21  
16:47:59 +05'30'

## DATA STRUCTURE PRACTICAL EXAM.

Learning gives creativity .  
creativity leads to thinking .  
Thinking provides you  
knowledge.  
Knowledge makes you great.



NIKHIL NK

Name - NIKHIL GAJANAN CHALIKWAR

Roll no - 12      Class - sycse1

Subject - Data structure lab.

•PRACTICAL INDEX•

SR.	NAME OF EXPERIMENT	DATE	PAGE	SIGN.
1	Oparations on Array.			
2	Sparse matrix.			
3	Binary search.			
4	Insertion sort.			
5	Selection sort.			
6	Radix sort			
7	Singly linked list.			
8	Doubly linked list.			
9	Stack.			
10	Infix to postfix			
11	Queue.			
12	Binary search tree.			



NIKHIL CHALIKWAR NK  
SYCSE1 S1 ROLL NO:-12

```
//ARRAY SORT
#include<stdio.h>
#include<conio.h>
void main()
{
int j,i,key,a[20],n,p;
clrscr();
printf("enter the no element");
scanf("%d",&n);
printf("enter the element");
for(p=0;p<=n;p++)
{
scanf("%d",&a[p]);
}
for(j=1;j<=n;j++)
{
key=a[j];
i=j-1;
while(i>0&&a[i]>key)
{
a[i+1]=a[i];
i=i-1;
}
a[i+1]=key;
}
for(i=1;i<=n;i++)
{
printf("\nsorted array is%d",a[i]);
}
```

```
}
```

```
getch();
```

```
}
```

**NIKHIL CHALIKWAR NK**  
**SYCSE1 S1 ROLL NO:-12**  
**//OPRATON ON ARRAY**  
+++++  
**#include<stdio.h>**  
**#include<conio.h>**  
**int a[20],ch,pos,i,b,n;**  
**void create();**  
**void insert();**  
**void del();**  
**void display();**  
**void main()**  
{  
clrscr();  
do  
{  
printf("1.create\n");  
printf("2.insert\n");  
printf("3.del\n");  
printf("4.display\n");  
printf("\nenter your choice");  
scanf("%d",&ch);  
switch(ch)  
{  
case 1:  
create();  
break;  
case 2:  
insert();  
}

```
break ;
case 3:
del();
break;
case 4:
display();
break;
default:
printf("invalid choice");
break;
}
while(ch!=4);
getch();
}
void create()
{
printf("enter the number of element of array");
scanf("%d",&n);
printf("enter element");
for(i=0;i<n;i++)
{
scanf("%d",&a[i]);
}
void insert()
{
printf("enter postion and element");
scanf("%d %d",&pos,&b);
for(i=n-1;i>=pos-1;i--)
{
a[i+1]=a[i];
```

```
}

a[pos-1]=b;
n=n+1;
}

void del()
{
printf("enter postion");
scanf("%d",&pos);
b=a[pos];
for(i=pos;i<n-1;i++)
{
a[i]=a[i+1];
}
n=n-1;
}
void display()
{
int i;
printf("\ndisplay array");
for(i=0;i<n;i++)
{
printf("\n%d",a[i]);
}
}
/*=====output=====
1.create
2.insert
3.del
4.display
enter your choice 1
enter the number of element of array 4
```

**enter element 4 6 7 3**

**1.create**

**2.insert**

**3.del**

**4.display**

**enter your choice 4**

**display array**

**4 6 7**

**3**

**===== \*/**

**+++++\*/**

**NIKHIL CHALIKWAR NK**  
**SYCSE1 S1 ROLL NO:-12**

```
//SLECTION SORT
#include<stdio.h>
#include<conio.h>
void swap();
int i,j,n,a[30],min,temp;
void main()
{
clrscr();
printf("Enter the no. of elements to be added\n");
scanf("%d",&n);
printf("Enter the elements\n");
//Create Array
for(i=0;i<n;i++)
{
scanf("%d",&a[i]);
}
//Selection Logic
for(i=0;i<n-1;i++)
{
min=i;
for(j=i+1;j<n;j++)
{
if(a[j]<a[i])
min=j;
}
swap();
}
printf("The sorted elements are : \n");
```

```
for(i=0;i<n;i++)
{
printf("%d\n",a[i]);
}
getch();
}

//Swapping Logic
void swap()
{
temp=a[i];
a[i]=a[min];
a[min]=temp;
}
/*=====Output=====
Enter the no. of elements to be added
5
Enter the elements
5 4 3 2 1
The sorted elements are :
1 2 3 4 5
=====*/
++++++
```

NIKHIL CHALIKWAR NK

SYCSE1 S1 ROLL NO:-12

//STACK PUSHPA

```
#####
#include<stdio.h>
#include<conio.h>
int stack[5];
int top=-1;
void push()
{
    int x;
    clrscr();
    printf("\n enter a number to push");
    scanf("%d",&x);
    top= top+1;
    stack[top]=x;
}
void pop()
{
    int y;
    y=stack[top];
    top--;
}
void display()
{
    int i;
    clrscr();
    for(i=0;i<5;i++)
    printf("\n stack is %d",stack[i]);
}
void main()
{
```

```
int s;
clrscr();
printf("enter 1 for push, enter 2 for pop: ");
scanf("%d",&s);
switch(s)
{
case 1:
push();
display();
break;
case 2:
pop();
break;
display();
}
getch();
}
/* Output:
enter 1 for push, enter 2 for pop: 1
enter a number to push12
12
0 0 0 0
*/
#####
#####
```

# NIKHIL CHALIKWAR

## SYCSE 1 S1 ROLL NO-12

```
//ARRAY SORT
#include<stdio.h>
#include<conio.h>
void swap();
int i,j,n,a[30],min,temp;
void main()
{
clrscr();
printf("Enter the no. of elements to be
added\n");
scanf("%d",&n);
printf("Enter the elements\n");
//Create Array
for(i=0;i<n;i++)
{
scanf("%d",&a[i]);
}
//Selection Logic
for(i=0;i<n-1;i++)
{
min=i;
for(j=i+1;j<n;j++)
{
if(a[j]<a[i])
min=j;
}
swap();
}
printf("The sorted elements are : \n");
```

```
for(i=0;i<n;i++)
{
printf("%d\n",a[i]);
}
getch();
}
//Swapping Logic
void swap()
{
temp=a[i];
a[i]=a[min];
a[min]=temp;
}
/*=====Output=====
=====
Enter the no. of elements to be added
5
Enter the elements
5 4 3 2 1
The sorted elements are :
1 2
3 4 5
=====
=*/

```

NIKHIL CHALIKWAR NK

SYCSE1 S1 ROLL NO:-12

//POSTFIX

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<string.h>
int stack[50];
char a[10];
int top=-1;
void push(int);
void cal(char);
void main()
{
int i;
clrscr();
printf("\n enter postfix expression");
gets(a);
for(i=0;i<strlen(a) ;i++)
{
if(a[i]>'0' && a[i]<'9')
{
push(a[i]);
}
if(a[i]=='+ || a[i]=='- ')
{
cal(a[i]);
}
printf("\n result=%d",stack[top]);
getch();
}
void push(int v)
```

```
{  
top++;  
stack[top]=(int)(a[v]-48);  
}  
void cal(char c)  
{  
int a,b,ans;  
b=stack[top];  
stack[top]='/0';  
top--;  
a=stack[top];  
stack[top]='/0';  
top--;  
switch(c)  
{  
case '+':  
ans=a+b;  
break;  
case '-':  
ans=a-b;  
break;  
} top++;  
stack[top]=ans;  
}  
// enter postfix expression3 7 +  
// result=10  
+++++  
+++++
```

**NIKHIL CHALIKWAD****S1 ROLL NO:->12**

```
/*Sparse Matrix*/  
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
/*intialization */  
int a[20][20];  
int x,y,o,i,j;  
int c=0,v=0;  
clrscr();  
/*input of no of rows and colummns*/  
printf("Enter no of rows");  
scanf("%d",&x);  
printf("\n Enter no of columns");  
scanf("%d",&y);  
/*input for elements of the matrix */  
o=x*y;  
for(i=0;i<x;i++)  
{  
for(j=0;j<y;j++)  
{  
scanf("%d",&a[i][j]);  
/* calculation of zero and non zero elements*/
```

```
if(a[i][j]==0)
{
c=c+1;
}
else
v=v+1;
}

/*to check whether a matrix is sparse or not */
if (c>x/2)
printf("\n it is a sparse matrix");
else
printf("\n it is not a sparse matrix");
/*to print the matrix*/
for(i=0;i<x;i++)
{
printf("\n");
for(j=0;j<y;j++)
{
printf("%d\t",a[i][j]);
}
}
/*to print the sparse matrix in row ,col,value form.*/
printf("\n");
```

```
printf("row\t");
printf("col\t");
printf("value\t");
printf("\n");
printf("%d \t %d \t %d \t",x,y,v);
for(i=0;i<x;i++)
{
    for(j=0;j<y;j++)
    {
        if(a[i][j]!=0)
        {
            printf("\n");
            printf("%d\t",i);
            printf("%d\t",j);
            printf("%d\t",a[i][j]);
        }
    }
}
getch();
}

/* output:Enter no of rows3
   Enter no of columns3
1
2
```

3  
0  
0  
0  
0  
0  
9

**it is a sparse matrix**

1 2 3  
0 0 0  
0 0 9

**row col value**

3	3	4
0	0	1
0	1	2
0	2	3
2	2	9

 \*/

**NIKHIL CHALIKWAR**  
**S1 ROLL NO:->12**

```
/*Singly Linked List*/  
  
#include<stdio.h>  
  
#include<conio.h>  
  
#include<stdlib.h>  
  
struct node  
{  
    int info;  
    struct node *next;  
};  
  
typedef struct node *NODEPTR;  
  
NODEPTR list=NULL;  
  
void ins_beg(ele)  
{  
    NODEPTR q;  
    q=(struct node*)malloc(sizeof(struct node));  
    q->info=ele;  
    q->next=list;  
    list=q;  
}  
  
void ins_end(ele)  
{
```

```
NODEPTR p,q;  
q=(struct node*)malloc(sizeof(struct node));  
q->info=ele;  
q->next=NULL;  
if(list==NULL)  
    ins_beg(ele);  
else  
{  
p=list;  
while(p->next!=NULL)  
{  
p=p->next;  
}  
p->next=q;  
}  
}
```

```
int count(void)  
{  
NODEPTR q;  
int count=1;  
q=list;
```

```
while(q->next!=NULL)
{
    q=q->next;
    count++;
}
return(count);
}

void ins_any(ele,pstn)
{
NODEPTR p,q;
int k,i;
q=(struct node*)malloc(sizeof(struct node));
q->info=ele;
q->next=NULL;
k=count();
if(pstn<=k+1)
{
    if(pstn==1)
        ins_beg(ele);
    else if(pstn==k+1)
        ins_end(ele);
    else
}
```

```
{  
p=list;  
for(i=0;i<pstn-2;i++)  
{  
p=p->next;  
}  
q->next=p->next;  
p->next=q;  
}  
}  
}
```

```
void del_beg()  
{  
NODEPTR q;  
if(list==NULL)  
printf("cannot delete the element ");  
else  
{  
q=list;  
list=q->next;  
free(q);
```

```
}

}

void del_end()

{

NODEPTR p,q;

p=q=list;

if(list==NULL)

    printf("underflow condition ");

else if(list->next==NULL)

{

    q=list;

    free(q);

    p=list;

    list=NULL;

}

else

{

    while(q->next!=NULL)

    {

        p=q;

        q=q->next;

    }

}
```

```
p->next=NULL;  
free(q);  
}  
}  
  
void display()  
{  
NODEPTR p;  
p=list;  
while(p!=NULL)  
{  
printf("-->[%d(%u) | %d]",p->info,p,p->next);  
p=p->next;  
}  
printf("\n");  
}  
  
int ele,pstn;  
void main()  
{  
int ch;  
clrscr();  
do  
{
```

```
printf("\t=====MENU=====\n");
printf("\t 1]insert from begining \n");
printf("\t 2]insert at the end \n");
printf("\t 3]insert at the any place \n");
printf("\t 4]delete from begining \n");
printf("\t 5]delete from end \n ");
printf("\t 6]display\n ");
printf("\t 0]exit the programm\n ");
printf("enter your choice ");
scanf("%d",&ch);
switch(ch)
{
    case 1:
        printf("enter the element you want to insert\n");
        scanf("%d",&ele);
        ins_beg(ele);
        break;
    case 2:
        printf("\n enter the element u want to insert at the end");
        scanf("%d",&ele);
        ins_end(ele);
        break;
}
```

**case 3:**

```
printf("\n enter the element you want to insert");
scanf("%d",&ele);
printf("\n enter the position ");
scanf("%d",&pstn);
ins_any(ele,pstn);
break;
```

**case 4:**

```
del_beg();
break;
```

**case 5:**

```
del_end();
break;
```

**case 6:**

```
display();
break;
}
}
while(ch);
getch();
}
/*OUTPUT:
```

**enter your choice 3**

**enter the element you want to insert2**

**enter the position 1**

=====MENU=====

- 1]insert from begning**
- 2]insert at the end**
- 3]insert at the any place**
- 4]delete from begning**
- 5]delete from end**
- 6]display**
- 0]exit the programm**

**enter your choice 6**

-->[2(2294) | 2270]-->[1(2270) | 2278]-->[3(2278) | 0]

=====MENU=====

- 1]insert from begning**
- 2]insert at the end**
- 3]insert at the any place**
- 4]delete from begning**
- 5]delete from end**
- 6]display**

0]exit the programm

enter your choice 0 \* /

**NIKHIL CHALIKWAN**  
**S1 ROLL NO:->12**

```
/*Radix Sort*/  
  
#include<stdio.h>  
  
#include<conio.h>  
  
int a[100],b[100][10],n,i,j,k,e,max;  
  
void radix(int a[],int n,int e)  
{  
  
for(i=0;i<k;i++)  
{  
    for(j=0;j<10;j++)  
    {  
        b[i][j]=0;  
    }  
}  
  
for(i=0;i<k;i++)  
{  
    n=(a[i]/e)%10;  
    b[i][n]=a[i];  
}  
  
for(i=0;i<k;i++)
```

```
{ printf("\n");
for(j=0;j<10;j++)
{ printf("%d",b[i][j]);
}
printf("\n");
for(i=0;i<10;i++)
{
for(j=0;j<k;j++)
{
if(b[j][i]!=0)
{
printf("\n%d",b[j][i]);
a[j]=b[j][i];
}
}
}
}

void main()
{
```

```
clrscr();
printf("enter size of array");
scanf("%d",&k);
printf("enter array elements");
for(i=0;i<k;i++)
{
    scanf("%d",&a[i]);
}
max=99;
for(e=1;max/e>0;e=e*10)
{
    radix(a,k,e);
}
getch();
}
/*=====output=====
enter size of array3
enter array elements12
43
25
```

**00120000000**

**00043000000**

**00000250000**

**12**

**43**

**25**

**01200000000**

**00004300000**

**00250000000**

**12**

**25**

**43**

**=====\*/**

**NIKHIL CHALIKWAR**  
**S1 ROLL NO:->12**

```
/*Doubly Linked List*/
#include<stdio.h>
#include<conio.h>
struct node{
    int data;
    struct node *prev;
    struct node *next;
};
struct node *head;
int a,i,x,p,d,m;
//to insert node at begining
void i1()
{
    struct node*temp;
    temp=(struct node*)malloc(sizeof(struct node));
    printf("enter node data");
    scanf("%d",&temp->data);
    if(head==NULL)
    {
        head=temp;
```

```
}

else

{
    head->prev=temp;
    temp->next=head;
    head=temp;
}

}

//insert node at end.

void i2()

{
    struct node*temp;
    struct node*temp1;
    temp=(struct node*)malloc(sizeof(struct node));
    printf("enter node data");
    scanf("%d",&m);
    temp->data=m;
    temp->next=NULL;
    if(head==NULL)
    {
        head=temp;
```

```
}

else

{

    temp1=head;

    while(temp1->next!=NULL)

    {

        temp1=temp1->next;

    }

    temp1->next=temp;

    temp->prev=temp1;

}

//insert node at nth position

void i3()

{

    struct node*temp;

    printf("enter element and position");

    scanf("%d %d",&x,&p);

    temp=(struct node*)malloc(sizeof(struct node));

    temp->data=x;

    temp->next=NULL;
```

```
if(p==1)
{
    temp->next=head;
    head=temp;
}

else
{
    struct node*temp1=head;
    struct node*temp2;
    for(i=0;i<p-1;i++)
    {
        temp2=temp1;
        temp1=temp1->next;
    }
    temp->prev=temp2;
    temp2->next=temp;

    temp->next=temp1;
    temp1->prev=temp;
}
}

/*

```

```
//deletion of node  
void d1()  
{  
    struct node *temp,*temp1,*temp2;  
    if(head==NULL)  
    {  
        printf("empty list");  
    }  
    else if(head->next==NULL)  
    {  
        temp=head;  
        free(temp);  
    }  
  
    else  
    {  
        temp=head;  
        head=temp->next;  
        head->prev=NULL;  
        free(temp);  
    }  
}
```

```
void d2()
{
    struct node *temp,*temp1,*temp2;
    if(head==NULL)
    {
        printf("empty list");
    }
    else if(head->next==NULL)
    {
        temp=head;
        free(temp);
    }

else
{
    temp1=head;
    while(temp1->next!=NULL)
    {
        temp=temp1;
        temp1=temp1->next;
    }
}
```

```
temp->next=NULL;  
  
free(temp1);  
}}  
*/  
void d3()  
{  
    struct node*temp,*temp1,*temp2;  
    printf("position at which element to be deleted");  
    scanf("%d",&d);  
    if(head==NULL)  
    {  
        printf("empty list");  
    }  
    else if(d==1)  
    {  
        temp=head;  
        head=temp->next;  
        free(temp);  
    }  
    else
```

```
{ temp=head;
for(i=0;i<=d-2;i++)
{
    temp=temp->next;
}
temp1=temp->next;
temp2=temp->prev;
temp1->prev=temp2;
temp2->next=temp1;
free(temp);
}

// display linked list
void display()
{
    struct node *temp=head;
    printf("data\taddress\n");
    while(temp!=NULL)
    {
        printf("%u<-%t%d\t%u->\n",temp->prev,temp->data,temp->next);
        temp=temp->next;
    }
}
```

```
}
```

```
void main()
```

```
{
```

```
    struct node *head=NULL;
```

```
    int ch;
```

```
    clrscr();
```

```
do{
```

```
    printf("\n 1 insert at begining");
```

```
    printf("\t 2insert at end");
```

```
    printf("\t 3 insert at nth pos");
```

```
    printf("\t 4 delete from beigning");
```

```
    printf("\t 5 delete from end");
```

```
    printf("\t 6 delete from nth position");
```

```
    printf("\t 7 display");
```

```
    printf("\t 8 exit");
```

```
    printf("\n enter choice");
```

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

```
switch(ch)
```

{

**case 1:****i1();**  
**break;****case 2:****i2();**  
**break;****case 3:****i3();**  
**break;**  
**/\*****case 4:****d1();**  
**break;****case 5:****d2();**

```
break; */
```

**case 6:**

```
d3();
```

```
break;
```

**case 7:**

```
display();
```

```
break;
```

**case 8:**

```
exit(0);
```

```
break;
```

**default:**

```
printf("invalid choice");
```

```
}
```

```
}while(ch!=8);
```

```
getch();
```

```
}
```

```
/*=====output=====
```

```
o/p:1
```

**1 insert at begining**

**2insert at end**

**3 insert at nth pos**

**4 delete from nth position**

**5 display**

**6 exit**

**enter choice1**

**enter node data23**

**1 insert at begining**

**2insert at end**

**3 insert at nth pos**

**4 delete from nth position**

**5 display**

**6 exit**

**enter choice2**

**enter node data24**

**1 insert at begining**

**2insert at end**

**3 insert at nth pos**

**4 delete from nth position**

**5 display**

**6 exit**

**enter choice3**

**enter element and position21 2**

**1 insert at begining**

**2insert at end**

**3 insert at nth pos**

**4 delete from nth position**

**5 display**

**6 exit**

**enter choice5**

**data    adres**

**23    2146-> 21    2138-> 24    0**

**1 insert at begining**

**2insert at end**

**3 insert at nth pos**

**4 delete from nth position**

**5 display**

**6 exit**

**enter choice6**

**o/p:2**

**1 insert at begining**

**2insert at end**

**3 insert at nth pos**

**4 delete from nth position**

**5 display**

**6 exit**

**enter choice1**

**enter node data22**

**1 insert at begining**

**2insert at end**

**3 insert at nth pos**

**4 delete from nth position**

**5 display**

**6 exit**

**enter choice2**

**enter node data33**

**1 insert at begining**

**2insert at end**

**3 insert at nth pos**

**4 delete from nth position**

**5 display**

**6 exit**

**enter choice4**

**position at which element to be deleted2**

**1 insert at begining**

**2insert at end**

**3 insert at nth pos**

**4 delete from nth position**

**5 display**

**6 exit**

**enter choice5**

**data    adres**

**22    0->**

**1 insert at begining**

**2insert at end**

**3 insert at nth pos**

**4 delete from nth position**

**5 display**

**6 exit**

**enter choice6\*/**

**NIKHIL CHALIKWAR**  
**S1 ROLL NO:->12**

```
/*Binary Search*/  
  
include<stdio.h>  
  
#include<conio.h>  
  
void main()  
{  
int n,low,high,mid,key,a[10],i,j;  
clrscr();  
printf("Enter the size of arrey");  
scanf("%d",&n);  
for(i=0;i<n;i++)  
{  
printf("\n Enter the elements of arrey");  
scanf("%d",&a[i]);  
}  
printf("\n Enter the number which is to be searched");  
scanf("%d",&key);  
low=0;
```

```
high=n-1;  
while(low<=high)  
{  
    mid=(low+high)/2;  
    if(key==a[mid])  
    {  
        printf("element is at %d",mid+1);  
        break;  
    }  
  
    else  
        if (key>a[mid])  
            low=mid+1;  
        else  
            high=mid-1;  
    }  
if(low>high)  
{  
    printf("\n Element nor found");
```

}

**getch();**

}

**/\* output**

**Enter the size of arrey4**

**Enter the elements of arrey34**

**Enter the elements of arrey4**

**Enter the elements of arrey5**

**Enter the elements of arrey56**

**Enter the number which is to be searched5**

**element is at 3      \*/**

**NIKHIL CHALIKWAN**  
**S1 ROLL NO:->12**

```
//AIM=>queue  
#include<stdio.h>  
#include<conio.h>  
#include<stdlib.h>  
#define MAX 5
```

```
int num, A[MAX], rear=-1, front=-1,i;
```

```
void enqueue(int num)  
{  
    if(rear==MAX-1)  
        printf("Overflow Condition\n");  
    else  
    {  
        ++rear;  
        A[rear]=num;  
        printf("Enqueued %d in A[%d]\n", num,rear);  
    }  
}
```

}

```
void dequeue()
{
    if(front== -1 && rear== -1)
    {
        printf("UnderFlow Condition.\n");
    }
    else if(front<=rear)
    {
        int temp;
        if(front== -1)
        {
            ++front;
        }
        temp = A[front];
        ++front;

        printf("Successfully Dequeued %d.\n", temp);
        A[front-1]=0;
    }
}
```

```
}

else

{

printf("Cannot perform dequeue further.");

}

}

void display()

{

int i;

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

{

printf("%d",A[i]);

}

}

void main()

{

int ch;

clrscr();
```

```
printf("====MENU====\n");
printf("\t[1] Enqueue\n\t[2] Dequeue\n\t[3] Display\n\t[4]
EXIT\n");
start:
printf("Enter your choice: ");
scanf("%d", &ch);
switch(ch)
{
    case 1:
        printf("Enter a number to enqueue: ");
        scanf("%d", &num);
        enqueue(num);
        goto start;
    case 2:
        dequeue();
        goto start;
    case 3:
        display();
        goto start;
```

**case 4:**

**exit(0);**

**default:**

**printf("Please enter a valid choice.");**

**goto start;**

**}**

**}**

**/\*=====OUTPUT=====**  
**=====**

**====MENU====**

**[1] Enqueue**

**[2] Dequeue**

**[3] Display**

**[4] EXIT**

**Enter your choice: 1**

**Enter a number to enqueue: 1**

**Enqueued 1 in A[0]**

**Enter your choice: 1**

**Enter a number to enqueue: 2**

**Enqueued 2 in A[1]**

**Enter your choice: 1**

**Enter a number to enqueue: 3**

**Enqueued 3 in A[2]**

**Enter your choice: 1**

**Enter a number to enqueue: 4**

**Enqueued 4 in A[3]**

**Enter your choice: 3**

**12340Enter your choice: 2**

**Successfully Dequeued 1.**

**Enter your choice: 3**

**02340Enter your choice:**

**4=====\*/**

**NIKHIL CHALIKWÄR**  
**S1 ROLL NO:->12**

```
#include<stdio.h>

#include<conio.h>

struct node

{

    struct node *left;

    int data;

    struct node *right;

};

struct node *root;

struct node* newnode(int ele)

{

    struct node *temp=(struct node*)malloc(sizeof(struct node));

    temp->left=NULL;

    temp->right=NULL;

    temp->data=ele;

    return (temp);

}

void Preorder(struct node *Node)
```

```
{  
if(Node==NULL)  
return;  
printf(" %d ", Node->data);  
Preorder(Node->left);  
Preorder(Node->right);  
}  
  
void Inorder(struct node *Node)  
{  
if(Node==NULL)  
return;  
Inorder(Node->left);  
printf(" %d ", Node->data);  
Inorder(Node->right);  
}  
  
void Postorder(struct node *Node)  
{  
if(Node==NULL)  
return;  
Postorder(Node->left);
```

```
Postorder(Node->right);

printf(" %d ", Node->data);

}

void main()

{

    struct node *root=newnode(1);

    clrscr();

    root->left=newnode(2);

    root->right=newnode(3);

    root->left->left=newnode(4);

    root->left->right=newnode(5);

    printf("\nPreorder Traversal of Binary Tree is: ");

    Preorder(root);

    printf("\nInorder Traversal of Binary Tree: ");

    Inorder(root);

    printf("\nPostorder Traversal of Binary Tree: ");

    Postorder(root);

    getch();

}
```

/\*=====OUTPUT=====

**Preorder Traversal of Binary Tree is: 1 2 4 5 3**

**Inorder Traversal of Binary Tree: 4 2 5 1 3**

**Postorder Traversal of Binary Tree: 4 5 2 3 1**

=====\*/

Thank you

- nk