ASSIGNMENT-1

Q1.Write a program to insert elements between an array.

CODE:

```
#include<conio.h>
#include<stdio.h>
void main()
{
int a[20],i,val,pos,s;
clrscr();
printf("\n Enter the size of an array: ");
scanf("%d",&s);
printf("Enter elements in an array: ");
for(i=0;i<s;i++)
scanf("%d",&a[i]);
printf("Enter position of the element where u want insert: ");
scanf("%d",&pos);
printf("Enter the value which u want to insert at that position: ");
scanf("%d",&val);
for(i=s;i>=pos;i--)
a[i]=a[i-1];
a[pos-1]=val;
s++;
for(i=0;i<s;i++)
printf("%d",a[i]);
getch();
}
```

OUTPUT:

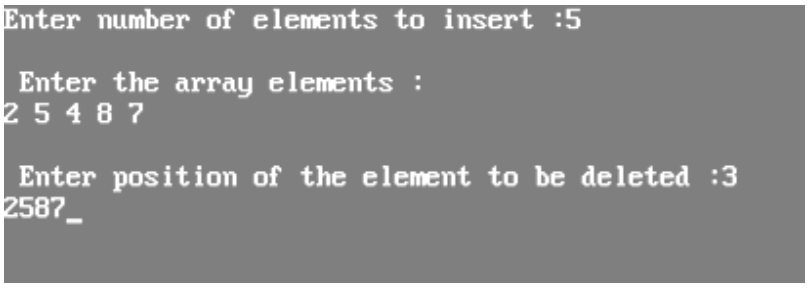
```
Enter the size of an array: 5
Enter elements in an array: 9 8 7 6 5
Enter position of the element where u want insert: 3
Enter the value which u want to insert at that position: 4
984765
```

Q2.Write a program to delete elements from an array.

CODE:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int arr[10],n,i,pos;
clrscr();
printf("Enter number of elements to insert :");
scanf("%d",&n);
printf("\n Enter the array elements :\n");
for(i=0;i<n;i++)
        scanf("%d",&arr[i]);
printf("\n Enter position of the element to be deleted :");
scanf("%d",&pos);
for(i=pos-1;i<n-1;i++)
        arr[i]=arr[i+1];
n=n-1;
for(i=0;i<n;i++)
        printf("%d",arr[i]);
getch();
}
```

OUTPUT:

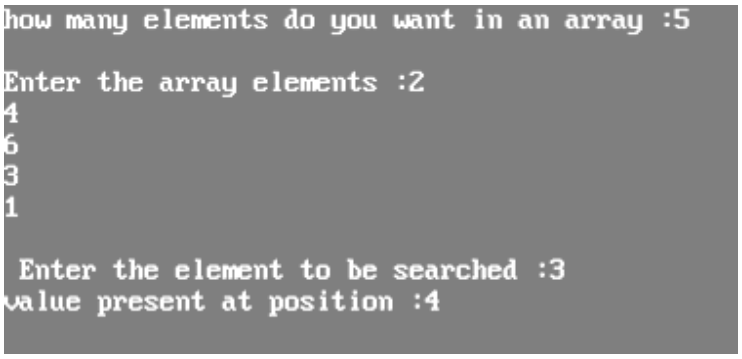


Q 3. Write a program to perform linear search.

CODE:

```
#include<stdio.h>
#include<conio.h>
void main()
{int arr[10],n,val,pos=-1,i;
clrscr();
printf("how many elements do you want in an array :");
scanf("%d",&n);
printf("\nEnter the array elements :");
for(i=0;i<n;i++)
scanf("%d",&arr[i]);
printf("\n Enter the element to be searched :");
scanf("%d",&val);
for(i=0;i<n;i++)
{if(arr[i]==val)
        {       pos=i;
                printf("value present at position :%d",pos+1);
        }
}
if(pos==-1)
printf("value not found!!!!!!!!!!");
getch();
}
```

OUTPUT:

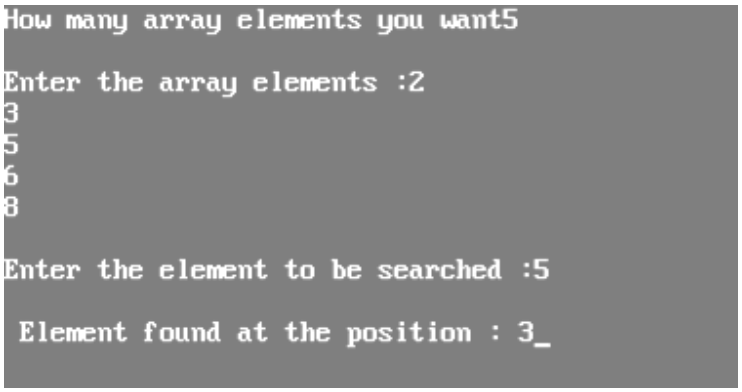


Q4.Write a program to perform binary search.

CODE:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int arr[10],n,i,pos;
clrscr();
printf("Enter number of elements to insert :");
scanf("%d",&n);
printf("\n Enter the array elements :\n");
for(i=0;i<n;i++)
        scanf("%d",&arr[i]);
printf("\n Enter position of the element to be deleted :");
scanf("%d",&pos);
for(i=pos-1;i<n-1;i++)
        arr[i]=arr[i+1];
n=n-1;
for(i=0;i<n;i++)
        printf("%d",arr[i]);
getch();
}
```

OUTPUT:



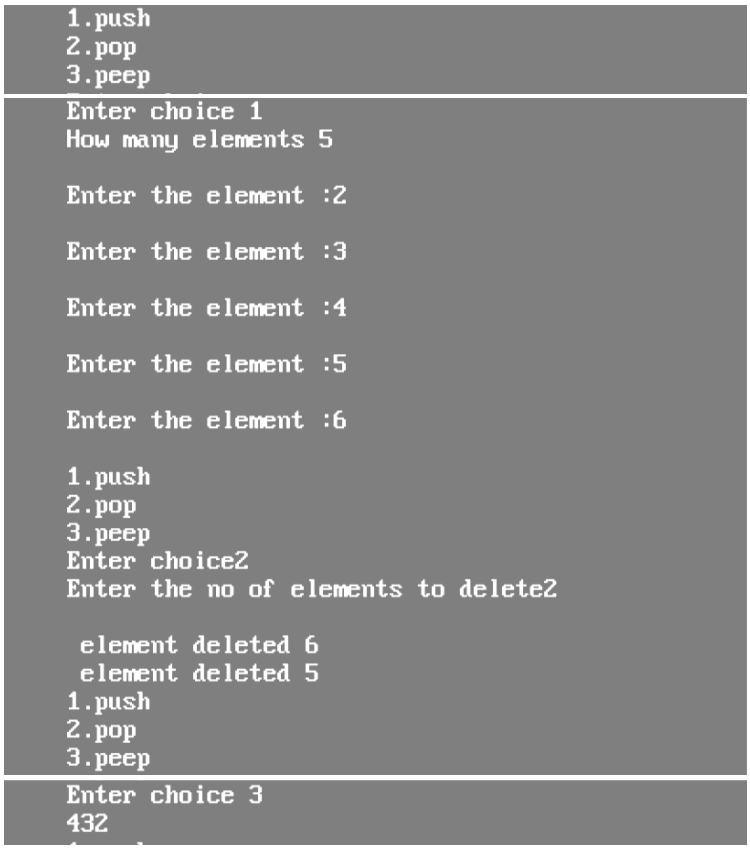
Q5.Write a program to perform push,pop,peep from a stack.

CODE:

```
#include<stdio.h>
#include<conio.h>
int st[5],top=-1,ele,i;
void push(int m);
void pop(int m);
void peep();
void main()
{
int ch,n;
clrscr();
do
{
printf("1.push \n2.pop \n3.peep\nEnter choice");
scanf("%d",&ch);
switch(ch)
{
case 1:printf("How many elements ");
scanf("%d",&n);
push(n);
break;
case 2: printf("Enter the no of elements to delete");
scanf("%d",&n);
pop(n);
break;
case 3:
peep();
break;
}
}while(ch!=0);
getch();
}
void push(int x)
{ if(top==4)
printf("\n overflow!!!!!!");
for(i=0;i<x;i++)
{
printf("\nEnter the element :");
scanf("%d",&ele)  ;
top++;
st[top]=ele;
}
}
void pop(int x)
```

```
{
if(top==-1 )
printf("\n underflow!!!!");
for(i=0;i<x;i++)
{
printf("\n element deleted %d",st[top]);
top--;
}
}
void peep()
{
for(i=top;i>=0;i--)
printf("%d",st[i]);
}
```

OUTPUT:



Q6 Write a program to add two sparse matrices.

CODE:

```
#include <stdio.h>
#include <conio.h>
void main()
{
        int sp1[10][3],sp2[10][3],sp3[10][3];
        clrscr();
        printf("\nEnter first sparse matrix");
        read_sp_mat(sp1);
        printf("\nEnter second sparse matrix");
        read_sp_mat(sp2);
        add_sp_mat(sp1,sp2,sp3);
        printf("\nFirst sparse matrix is");
        print_sp_mat(sp1);
        printf("\nSecond sparse matrix is");
        print_sp_mat(sp2);
        printf("\nThird sparse matrix is");
        print_sp_mat(sp3);
        getch();
}
int read_sp_mat(int sp[10][3])
{       int r,c,i,j,k,t;
        printf("\nEnter r and c : ");
        scanf("%d %d",&r,&c);
        printf("\nEnter the data \n");
        k=1;
        for(i=0;i < r;i++)
        {       for(j=0;j<c;j++)
                {
                        scanf("%d",&t);
```

```
                        if( t != 0 )
                        {
                                sp[k][0] = i;
                                sp[k][1] = j;
                                sp[k][2] = t;
                                k++;
                        }
                }
        }
        sp[0][0] = r;
        sp[0][1] = c;
        sp[0][2] = k-1;
        return;
}
int print_sp_mat(int sp[10][3])
{
        int r,c,i,j,tot_val,k;
        r = sp[0][0];
        c = sp[0][1];
        tot_val = sp[0][2];
        for(i=0;i<r;i++)
        {
                printf("\n");
                for(j=0;j<c;j++)
                {
                        for(k=1;k<=tot_val;k++)
                        {
                                if( sp[k][0] == i && sp[k][1] == j )
                                break;
                        }
                        if( k > tot_val)
                                printf("%4d",0);
                        else
```

```
                        printf("%4d",sp[k][2]);
                }
        }
        return;
}
int add_sp_mat(sp1,sp2,sp3)
int sp1[10][3],sp2[10][3],sp3[10][3];
{
        int r,c,i,j,k1,k2,k3,tot1,tot2;
        if( sp1[0][0] != sp2[0][0] || sp1[0][1] != sp2[0][1] )
        {
                printf("Invalid matrix size ");
                exit(0);
        }
        tot1 = sp1[0][2];
        tot2 = sp2[0][2];
        k1 = k2 = k3 = 1;
        while ( k1 <= tot1 && k2 <= tot2)
        {
                if ( sp1[k1][0] < sp2[k2][0] )
                {
                        sp3[k3][0] = sp1[k1][0];
                        sp3[k3][1] = sp1[k1][1];
                        sp3[k3][2] = sp1[k1][2];
                        k3++;k1++;
                }
                else
                if ( sp1[k1][0] > sp2[k2][0] )
                {
                        sp3[k3][0] = sp2[k2][0];
                        sp3[k3][1] = sp2[k2][1];
                        sp3[k3][2] = sp2[k2][2];
                        k3++;k2++;
```

```
            }
            else if ( sp1[k1][0] == sp2[k2][0] )
            {
            if ( sp1[k1][1] < sp2[k2][1] )
            {
                    sp3[k3][0] = sp1[k1][0];
                    sp3[k3][1] = sp1[k1][1];
                    sp3[k3][2] = sp1[k1][2];
                    k3++;k1++;
            }
            else
            if ( sp1[k1][1] > sp2[k2][1] )
            {
                    sp3[k3][0] = sp2[k2][0];
                    sp3[k3][1] = sp2[k2][1];
                    sp3[k3][2] = sp2[k2][2];
                    k3++;k2++;
            }
            else
            {
                    sp3[k3][0] = sp2[k2][0];
                    sp3[k3][1] = sp2[k2][1];
                    sp3[k3][2] = sp1[k1][2] + sp2[k2][2];
                    k3++;k2++;k1++;
            }
            }
    }
    while ( k1 <=tot1 )
    {
            sp3[k3][0] = sp1[k1][0];
            sp3[k3][1] = sp1[k1][1];
            sp3[k3][2] = sp1[k1][2];
            k3++;k1++;
```

```
        }

        while ( k2 <= tot2 )
        {
                sp3[k3][0] = sp2[k2][0];
                sp3[k3][1] = sp2[k2][1];
                sp3[k3][2] = sp2[k2][2];
                k3++;k2++;
        }
        sp3[0][0] = sp1[0][0];
        sp3[0][1] = sp1[0][1];
        sp3[0][2] = k3-1;
        return 0;
}
```

OUTPUT:

```
Enter first sparse matrix
Enter r and c : 2
2

Enter the data
1
0
2
0

Enter second sparse matrix
Enter r and c : 2
2
```

```
Enter the data
0
0
0
8

First sparse matrix is
   1   0
   2   0
Second sparse matrix is
   0   0
   0   8
Third sparse matrix is
   1   0
   2   8
```

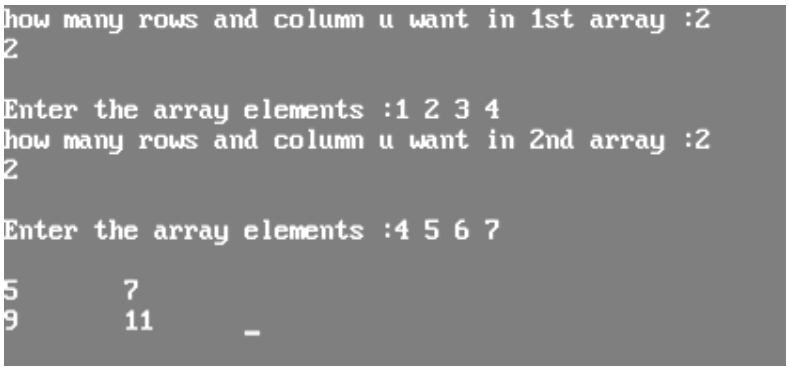
Q7. Write a program to add 2 2-D array.

CODE:

```
#include<conio.h>
#include<stdio.h>
void main()
{
int arr1[5][5],arr2[5][5],arr3[5][5],i,j ,n,m ,p,q;
clrscr();
printf("how many rows and column u want in 1st array :");
scanf("%d%d",&m,&n);
printf("\nEnter the array elements :");
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{
scanf("%d",&arr1[i][j]);
}}
printf("how many rows and column u want in 2nd array :");
scanf("%d%d",&p,&q);
printf("\nEnter the array elements :");
for(i=0;i<p;i++)
{
for(j=0;j<q;j++)
{
scanf("%d",&arr2[i][j]);
}}
if(m!=p&&n!=q)
printf("\n array cannot be added .....");
else
```

```
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{
arr3[i][j]=arr1[i][j]+arr2[i][j];
}}
for(i=0;i<m;i++)
{printf("\n");
for(j=0;j<n;j++)
{
printf("%d",arr3[i][j]);
printf("\t");
}}
getch();
}
```

OUTPUT:



ASSIGNMENT-2

Q1. Write a program to convert the infix expression to prefix expression.

CODE:

```
#define SIZE 50
#include<string.h>
#include <ctype.h>
#include<stdio.h>
char s[SIZE]; int top=-1;
push(char elem)
{
        s[++top]=elem;
}
char pop()
{
return(s[top--]);
}
int pr(char elem)
{
switch(elem)
{
case '#': return 0;
case ')': return 1;
case '+':
case '-': return 2;
case '*':
case '/':return 3;
}
}
void main()
{
char infx[50],prfx[50],ch,elem;
int i=0,k=0;
clrscr();
printf("\n\nRead the Infix Expression ? ");
scanf("%s",infx);
push('#');
strrev(infx);
while( (ch=infx[i++]) != '\0')
{
if( ch == ')')
```

```
push(ch);
else if(isalnum(ch))
prfx[k++]=ch;
else if( ch == '(')
{
while( s[top] != ')')
prfx[k++]=pop();
elem=pop();
}
else
{
while( pr(s[top]) >= pr(ch) )
prfx[k++]=pop(); push(ch);
}
}
while( s[top] != '#')
prfx[k++]=pop();
prfx[k]='\0';
strrev(prfx);
strrev(infx);
printf("\n\nGiven Infix Expn: %s \nPrefix Expn: %s\n",infx,prfx);
getch();
}
```

OUTPUT:

```
Read the Infix Expression :a*b%c(e^f*m)%d*s


Given Infix Expn: a*b%c(e^f*m)%d*s
Prefix Expn: %*ab%c*^efm*ds
```

Q2. Write a program to convert the infix expression to postfix.

CODE:

```
#include<stdio.h>
char stack[20];
int top = -1;
void push(char x)
{
   stack[++top] = x;
}

char pop()
{
   if(top == -1)
         return -1;
   else
         return stack[top--];
}

int priority(char x)
{
   if(x == '(')
         return 0;
   if(x == '+' || x == '-')
         return 1;
   if(x == '*' || x == '/')
         return 2;
}

void main()
{
   char exp[20];
   char *e, x;
   clrscr();
   printf("Enter the expression :: ");
   scanf("%s",exp);
   e = exp;
   while(*e != '\0')
   {
         if(isalnum(*e))
            printf("%c",*e);
         else if(*e == '(')
            push(*e);
```

```
        else if(*e == ')')
        {
           while((x = pop()) != '('
                 printf("%c", x);
        }
        else
        {
           while(priority(stack[top]) >= priority(*e))
                 printf("%c",pop());
           push(*e);
        }
        e++;
   }
   while(top != -1)
   {
        printf("%c",pop());
   }
   getch();
}
```

OUTPUT:

```
Enter the expression :: a*b^c(d^e*m)%s
abcde^m*^s%*_
```

ASSIGNMENT-3

Q1. Write a program to perform insert,delete and display operations on linear queue.

CODE:

```
#include<stdio.h>
#include<conio.h>
int q[5],f=-1,r=-1,i;
void ins(int );
void del(int);
void print();
void main()
{
int ch,n;
clrscr();
do
 { printf("\n1.insert\n2.delete\n3.print\nEnter choice:");
 scanf("%d",&ch);
 switch(ch)
 {
 case 1:printf("Enter the total elements u want to insert :");
          scanf("%d",&n);
          ins(n);
          break;
 case 2:printf("Enter the number of elements to delete :");
          scanf("%d",&n);
          del(n);
          break;
 case 3:print();
          break;

 }}while(ch!=0) ;
 getch();
 }
 void ins(int x)
 {
 int val;
 printf("Enter the value to be inserted");

 for(i=0;i<x;i++)
 {  scanf("%d",&val);
 if(r==4)
```

```
{
printf("\noverflow!!!!");
}
else if((f==-1)&&(r==-1))
{
f=0;
r=0;
}
else
r=r+1;
q[r]=val;
}
}
void del(int x)
{
for(i=0;i<x;i++)
{
if((f==-1)||(f==r+1))
{
printf("\nunderflow!!!!!!!");
}
else
f=f+1;
}}
void print()
{ printf("\nelements in the queue are :");
for(i=f;i<=r;i++)
{
printf("\n%d",q[i]);
}
}
```

OUTPUT:

```
1.insert
2.delete
3.print
```

```
Enter choice:1
Enter the total elements u want to insert :5
Enter the value to be inserted 1 4 6 8 5

1.insert
2.delete
3.print
Enter choice:2
Enter the number of elements to delete :2

1.insert
2.delete
3.print
Enter choice:3

elements in the queue are :
6
8
5
```

ASSIGNMENT-3

Q1. Write a program to perform insert,delete and display operations on linear queue.

CODE:

```
#include<stdio.h>
#include<conio.h>
int q[5],f=-1,r=-1,i;
void ins(int );
void del(int);
void print();
void main()
{
int ch,n;
clrscr();
do
 { printf("\n1.insert\n2.delete\n3.print\nEnter choice:");
 scanf("%d",&ch);
 switch(ch)
 {
 case 1:printf("Enter the total elements u want to insert :");
          scanf("%d",&n);
          ins(n);
          break;
 case 2:printf("Enter the number of elements to delete :");
          scanf("%d",&n);
          del(n);
          break;
 case 3:print();
          break;

 }}while(ch!=0) ;
 getch();
 }
 void ins(int x)
 {
 int val;
 printf("Enter the value to be inserted");

 for(i=0;i<x;i++)
 {  scanf("%d",&val);
 if(r==4)
 {
 printf("\noverflow!!!!");
 }
```

```
else if((f==-1)&&(r==-1))
{
f=0;
r=0;
}
else
r=r+1;
q[r]=val;
}
}
void del(int x)
{
for(i=0;i<x;i++)
{
if((f==-1)||(f==r+1))
{
printf("\nunderflow!!!!!!!");
}
else
f=f+1;
}}
void print()
{ printf("\nelements in the queue are :");
for(i=f;i<=r;i++)
{
printf("\n%d",q[i]);
}
}
```

OUTPUT:

```
1.insert
2.delete
3.print
Enter choice:1
Enter the total elements u want to insert :5
Enter the value to be inserted 1 4 6 8 5

1.insert
2.delete
3.print
Enter choice:2
Enter the number of elements to delete :2

1.insert
2.delete
3.print
Enter choice:3

elements in the queue are :
6
8
5
```

Q2. Write a program to insert,delete,and display operations on circular queue.

CODE:

```
#include<stdio.h>
#include<conio.h>
int a[5],r=-1,f=-1,i;
void ins(int);
void del(int);
void display();
void main()
{ int n,ch;
  clrscr();
  do
  {       printf("\n1:Insert");
          printf("\t2:Deletion");
          printf("\t3:Print");
          printf("\nEnter the choice");
          scanf("%d",&ch);
          switch(ch)
          {       case 1: printf("\nhow many elements");
                          scanf("%d",&n);
                          ins(n);
                          break;
                  case 2: printf("\nhow many elements");
                          scanf("%d",&n);
                          del(n);
                          break;
                  case 3: display();
                          break;
          }
  } while(ch!=4);
  getch();
}
void ins(int n)
{  int val ;
          printf("\n Enter the value to be inserted :\n ");
  for(i=0;i<n;i++)
  {         scanf("%d",&val);
          if(((f==0)&&(r==4))||(f==r+1))
          {       printf("\n overflow");
                  exit(0);
          }
          else if(r==-1)
          {         f=0;
                    r=0;
          }
```

```
        else if(r==4)
                r=0;
        else
                r=r+1;
  a[r]=val;
     }
  }
void del(int n)
{       for(i=0;i<n;i++)
        {       if(f==-1)
                {       printf("\n underflow");
                        exit(0);
                }
                if(f==r)
                {       f=-1,r=-1;  }
                else if(f==4)
                        f=0;
                else
                        f=f+1;
        }
}
void display()
{       printf("\n elements of the array");
        for(i=f;i<=r;i++)
        printf("\n%d",a[i]);
}
```

OUTPUT:

```
1:Insert        2:Deletion      3:Print
Enter the choice1

how many elements5

 Enter the value to be inserted :
 3 5 6 4 9

1:Insert        2:Deletion      3:Print
Enter the choice2

how many elements2

1:Insert        2:Deletion      3:Print
Enter the choice3

 elements of the array
6
4
9
```

Q3.Write a program to perform insert ,delete and display operations from double ended queue.

CODE:

```
#include<stdio.h>
#include<conio.h>
#define MAX 5
int deque[MAX];
int left=-1, right=-1,i;
void input_deque(void);
void output_deque(void);
void insert_right(void);
void insert_left(void);
void delete_right(void);
void delete_left(void);
void display(void);
int main()
{
 int ch;
 clrscr();

        printf("\n1:Input Restricted deque");
        printf("\t2:Output Restricted deque");
        printf("\t3:exit");
        printf("\n\nEnter your choice ");
        scanf("%d",&ch);
        switch(ch)
                {
                case 1: input_deque();
                        break;
                case 2: output_deque();
                        break;

                }
  getch();
 return 0;
}
void input_deque()
{
  int ch;
  do
  {
        printf("\n1.Insert right ");
        printf("\t2.Delete right ");
        printf("\t3.Delete left ");
        printf("\t4.Display ");
```

```
        printf("\t5.exit");
        printf("\nenter choice");
        scanf("%d",&ch);
        switch(ch)
        {
                case 1:insert_right();
                        break;
                case 2:delete_right();
                        break;
                case 3:delete_left();
                        break;
                case 4:display();
                        break;
        }
 }while(ch!=5);
}
void output_deque()
{
  int ch;
  do
  {
        printf("\n1.Insert right ");
        printf("\t2.Insert left ");
        printf("\t3.Delete left ");
        printf("\t4.Display ");
        printf("\t5.Exit");
        printf("\nenter choice");
        scanf("%d",&ch);
        switch(ch)
        {
                case 1:insert_right();
                        break;
                case 2:insert_left();
                        break;
                case 3:delete_left();
                        break;
                case 4:display();
                        break;
        }
 }while(ch!=5);
}


void insert_right()
{
 int n, val;
 printf("\nEnter the number of values to be added ");
 scanf("%d",&n);
```

```
 for(i=0;i<n;i++)
 {
 scanf("%d",&val);
 if( (left==0 && right==MAX-1) || (left==right+1) )
 {
          printf("\nOVERFLOW");
 }
 if(left==-1)
 {
          left=0;
          right=0;
 }
 else
 {
          if(right==MAX-1)
          right=0;
          else
          right=right+1;
 }
 deque[right]=val;
} }

void insert_left()
{
 int n,val;
 printf("\nEnter the number of values to be added ");
 scanf("%d",&n);
 for(i=0;i<n;i++)
{ scanf("%d",&val);
 if( (left==0 && right==MAX-1) || (left==right+1) )
 {
          printf("\nOVERFLOW");
 }
 if(left==-1)
 {
          left=0;
          right=0;
 }
 else
 {
          if(left==0)
                   left=MAX-1;
          else
                   left=left-1;
 }
 deque[left]=val;
}
}
```

```
void delete_right()
{
 if(left==-1)
 {
        printf("\nUNDERFLOW");
        return;
 }
 printf("\nThe deleted element is %d\n", deque[right]);
 if(left==right)
 {
        left=-1;
        right=-1;
 }
 else
 {
        if(right==0)
        right=MAX-1;
  else
        right=right-1;
 }
}


void delete_left()
{
 if(left==-1)
 {
        printf("\nUNDERFLOW");
        return;
 }
 printf("\nThe deleted element is %d\n", deque[left]);
 if(left==right)
 {
        left=-1;
        right=-1;
 }
 else
 {
        if(left==MAX-1)
        left=0;
  else
        left=left+1;
 }
}


void display()
{
```

```
int front=left, rear=right;
if(front==-1)
{
        printf("\nQueue is Empty\n");
        return;
}
printf("\nThe elements in the queue are: ");
if(front<=rear)
{
        while(front<=rear)
        {
                printf("%d\t",deque[front]);
                front++;
        }
}
else
{
        while(front<=MAX-1)
 {
                printf("%d\t",deque[front]);
                front++;
 }
        front=0;
         while(front<=rear)
 {
                printf("%d\t",deque[front]);
                 front++;
 }
 }
printf("\n");
}
```

OUTPUT:

```
1:Input Restricted deque        2:Output Restricted deque       3:exit

Enter your choice 1

1.Insert right  2.Delete right  3.Delete left   4.Display       5.exit
enter choice1

Enter the number of values to be added 5
1 2 3 4 5

1.Insert right  2.Delete right  3.Delete left   4.Display       5.exit
enter choice2

The deleted element is 5

1.Insert right  2.Delete right  3.Delete left   4.Display       5.exit
enter choice3

The deleted element is 1
```

```
1.Insert right  2.Delete right  3.Delete left   4.Display       5.exit
enter choice4

The elements in the queue are: 2        3       4
```

```
1:Input Restricted deque        2:Output Restricted deque       3:exit

Enter your choice 2

1.Insert right  2.Insert left   3.Delete left   4.Display       5.Exit
enter choice1

Enter the number of values to be added 2
1 3

1.Insert right  2.Insert left   3.Delete left   4.Display       5.Exit
enter choice2

Enter the number of values to be added 3
2 4 5

1.Insert right  2.Insert left   3.Delete left   4.Display       5.Exit
enter choice3

The deleted element is 5
```