Practical No:- 4.(c )

1. c Write a program to implement the concept of deque.

Practical Implementation:-

Code:-

#include <stdio.h>

#define size 5

**int** deque[size];

**int** f = -1, r = -1;

//  insert\_front function will insert the value from the front

**void** insert\_front(**int** x)

{

**if**((f==0 && r==size-1) || (f==r+1))

    {

        printf("Overflow");

    }

**else** **if**((f==-1) && (r==-1))

    {

        f=r=0;

        deque[f]=x;

    }

**else** **if**(f==0)

    {

        f=size-1;

        deque[f]=x;

    }

**else**

    {

        f=f-1;

        deque[f]=x;

    }

}

// insert\_rear function will insert the value from the rear

**void** insert\_rear(**int** x)

{

**if**((f==0 && r==size-1) || (f==r+1))

    {

        printf("Overflow");

    }

**else** **if**((f==-1) && (r==-1))

    {

        r=0;

        deque[r]=x;

    }

**else** **if**(r==size-1)

    {

        r=0;

        deque[r]=x;

    }

**else**

    {

        r++;

        deque[r]=x;

    }

}

// display function prints all the value of deque.

**void** display()

{

**int** i=f;

    printf("\nElements in a deque are: ");

**while**(i!=r)

    {

        printf("%d ",deque[i]);

        i=(i+1)%size;

    }

     printf("%d",deque[r]);

}

// getfront function retrieves the first value of the deque.

**void** getfront()

{

**if**((f==-1) && (r==-1))

    {

        printf("Deque is empty");

    }

**else**

    {

        printf("\nThe value of the element at front is: %d", deque[f]);

    }

}

// getrear function retrieves the last value of the deque.

**void** getrear()

{

**if**((f==-1) && (r==-1))

    {

        printf("Deque is empty");

    }

**else**

    {

        printf("\nThe value of the element at rear is %d", deque[r]);

    }

}

// delete\_front() function deletes the element from the front

**void** delete\_front()

{

**if**((f==-1) && (r==-1))

    {

        printf("Deque is empty");

    }

**else** **if**(f==r)

    {

        printf("\nThe deleted element is %d", deque[f]);

        f=-1;

        r=-1;

    }

**else** **if**(f==(size-1))

     {

         printf("\nThe deleted element is %d", deque[f]);

         f=0;

     }

**else**

     {

          printf("\nThe deleted element is %d", deque[f]);

          f=f+1;

     }

}

// delete\_rear() function deletes the element from the rear

**void** delete\_rear()

{

**if**((f==-1) && (r==-1))

    {

        printf("Deque is empty");

    }

**else** **if**(f==r)

    {

        printf("\nThe deleted element is %d", deque[r]);

        f=-1;

        r=-1;

    }

**else** **if**(r==0)

     {

         printf("\nThe deleted element is %d", deque[r]);

         r=size-1;

     }

**else**

     {

          printf("\nThe deleted element is %d", deque[r]);

          r=r-1;

     }

}

**int** main()

{

    insert\_front(20);

    insert\_front(10);

    insert\_rear(30);

    insert\_rear(50);

    insert\_rear(80);

    display();  // Calling the display function to retrieve the values of deque

    getfront();  // Retrieve the value at front-end

    getrear();  // Retrieve the value at rear-end

    delete\_front();

    delete\_rear();

    display(); // calling display function to retrieve values after deletion

**return** 0;

}

Output:-

