//Program to Implement De-queue using arrays

#include<stdio.h>

#include<stdlib.h>

int DQ[100],N,F=-1,R=-1,Item;

void InsertRearEnd()

{

  if(F==0 && R==N-1 || F==R+1) //Full Conditions of Deque

  {

    printf("Deque Overflow , No Element can be inserted\n");

    return; //returns the control back to the main

  }

   printf("\nEnter the Element to be inserted from Rear End\n");

       scanf("%d",&Item);

if( F==-1 )// if queue is empty & element is inserted for the first time

{

F=R=0;

}

  else if (R==N-1)

    R=0;

  else

    R++;

DQ[R] = Item;

  printf("Element %d has been inserted\n", Item);

}

void DeleteFromFront()

{

  if(F==-1)

  {

    printf("Deque Underflow , No Element is Present\n");

return;

  }

  printf("Element %d has been deleted from Front\n", DQ[F]);

  if(F==R) //only one element is present in Deque

   F=R=-1; //one element deleted , no element present

  else if(F==N-1)

    F=0;

  else

    F=F+1;

}

//Deletion from Rear & Insertion From Front End is the complete opposite of the usual way , only checking full conditions is the same.

void DeleteFromRear() //for deleting from rear end we do r--;

{

if(R==-1)

{ printf("Deque Underflow, No Element is Present\n");

return;}

printf("Element %d is deleted from Rear\n",DQ[R]);

if(F==R)

   F=R=-1;

else if(R==0) //since we are decrementing R , we use R==N if R==1.

   R=N-1;

else

   R--;

}

void InsertFrontEnd()

{ if(F==0 && R==N-1 || F==R+1)

  { printf("Deque Overflow , No Element can be inserted\n");

  return; }

  printf("Insert the Element to be inserted from Front End\n");

       scanf("%d",&Item);

if( F==-1 )// if queue is empty

{

F=R=0;

}

else if(F==0)

   F=N-1;

else

   F--;

DQ[F] = Item;

printf(" Element %d has been inserted \n" , Item);

}

void DisplayDQ()

  { int i;

   if(F==-1)

   { printf(" Deque Underflow , No Elements Present\n");

   return;}

   printf("Deque Is :\n");

    if(R>=F)

  {

    for(i=F;i<=R;i++)

      printf("%d \t" , DQ[i]);

  }

  else

  { for(i=F;i<N;i++)

    printf("%d \t" , DQ[i]);

  for(i=0; i<=R ;i++)

    printf("%d \t", DQ[i]);

  }

  printf("\n");

}

int main()

{

  int item;

  int ch=1;

  int choice=1;

  printf("Enter the Maximum Number Of Elements\n");

scanf("%d",&N);

  printf("Enter 1 For Input Restricted Deque / 2 For Output Restricted Deque\n");

  scanf("%d",&choice);

  while(ch>=1 && ch<=5)

  {

    printf("Enter Your Choice\n");

    printf(" 1. Insert From Rear End \n");

    printf(" 2. Delete From Front End\n");

    if(choice==2)

    printf(" 3. Insert From Front End\n");

   else

    printf(" 4. Delete From Rear End\n");

    printf(" 5. Display \n");

    printf(" 6. Exit \n");

    scanf("%d",&ch);

    switch(ch)

    {

      case 1 :

           InsertRearEnd();

           DisplayDQ();

           break;

      case 2 : DeleteFromFront();

       DisplayDQ();

       break;

      case 3 :

           InsertFrontEnd();

           DisplayDQ();

           break;

      case 4 : DeleteFromRear();

       DisplayDQ();

break;

      case 5 : DisplayDQ();

       break;

      case 6 : exit(0);

       break ;

default : printf("\nInvalid Choice\n");

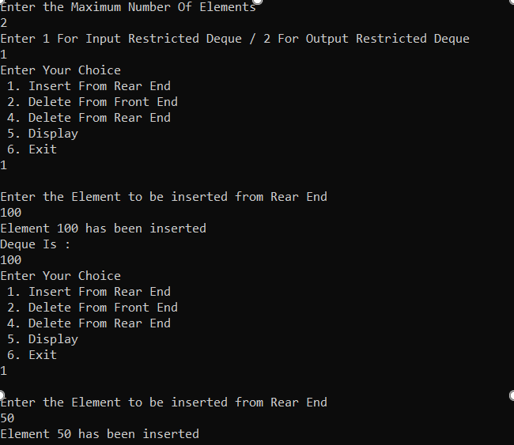
}

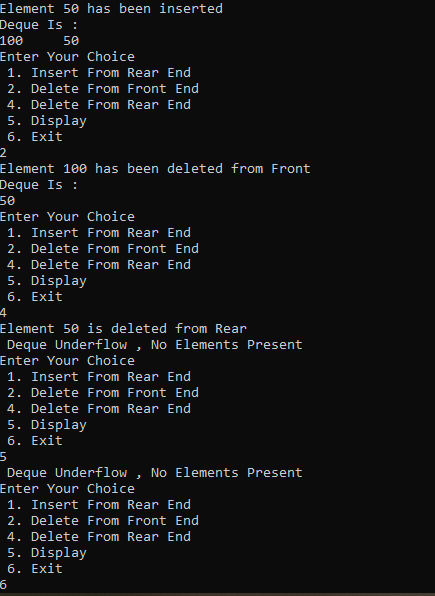
  } //end of while loop

}

Output :

1. Input Restricted Deque:





1. Output Restricted Deque:

