**DEPARTMENT OF INFORMATION TECHNOLOGY**

**Course Name and Code:** Data Structures Lab **(**ITL302)

**Semester:** III (SYIT)

**Academic Year:** 2024-25 (Odd Semester)

**Experiment No. 04**

**Aim:** **Array Implementation of Double-ended Queue using Array for real-world application.**

**Code:**

#include <stdio.h>

#include <stdlib.h>

struct Dequeue

{

int n;

int deq[1000];

int front;

int rear;

};

int isEmpty(struct Dequeue \* d)

{

if(d->front == -1 && d->rear == -1)

return 1;

return 0;

}

int isFull(struct Dequeue \* d)

{

if(d->front - d->rear == 1 || (d->front == 0 && d->rear == d->n-1))

return 1;

return 0;

}

void Display(struct Dequeue \* d)

{

int i;

printf("[");

if(d->front <= d->rear && !isEmpty(d))

{

for(i = d->front; i <= d->rear; i++)

printf("%d, ", d->deq[i]);

}

else if(d->front > d->rear)

{

for(i = d->front; i < d->n; i++)

printf("%d, ", d->deq[i]);

for(i = 0; i <= d->rear; i++)

printf("%d, ", d->deq[i]);

}

printf("]\n");

}

void Insert\_Front(struct Dequeue \* d, int ele)

{

if(isFull(d))

{

printf("Overflow error");

exit(-1);

}

if(isEmpty(d))

{

d->front = 1;

d->rear = 0;

}

else if(d->front == 0)

d->front = d->n;

d->front -= 1;

d->deq[d->front] = ele;

}

void Insert\_Rear(struct Dequeue \* d, int ele)

{

if(isFull(d))

{

printf("Overflow error");

exit(-1);

}

if(isEmpty(d))

d->front = 0;

else if(d->rear == d->n - 1)

d->rear = -1;

d->rear += 1;

d->deq[d->rear] = ele;

}

void Delete\_Front(struct Dequeue \* d)

{

if(d->front == -1)

{

printf("Underflow error");

exit(-1);

}

if(d->front == d->n - 1)

d->front = 0;

else if(d->front == d-> rear)

{

d->front = -1;

d->rear = -1;

}

else

d->front += 1;

}

void Delete\_Rear(struct Dequeue \* d)

{

if(d->rear == -1)

{

printf("Underflow error");

exit(-1);

}

if(d->rear == 0)

d->rear = d->n - 1;

else if(d->rear == d-> front)

{

d->rear = -1;

d->front = -1;

}

else

d->rear -= 1;

}

int main()

{

struct Dequeue de;

int rest, choice;

printf("Enter Dequeue length: ");

scanf("%d", &de.n);

printf("Choose type of dequeue: \n1.Input Restricted\n2.Output Restricted\n");

scanf("%d", &rest);

if(rest != 1 && rest != 2)

return -1;

de.front = -1;

de.rear = -1;

while(1)

{

int ele;

printf("Dequeue: \n");

Display(&de);

printf("Choose an option: \n0) Exit\n1) Input from Left\n2) Delete from Left\n3) ");

if(rest == 1)

printf("Delete from Right\n");

else

printf("Input from Right\n");

scanf("%d", &choice);

switch(choice)

{

case 1:

printf("Enter element to be inputted: ");

scanf("%d", &ele);

Insert\_Front(&de, ele);

break;

case 2:

Delete\_Front(&de);

break;

case 3:

if(rest == 1)

Delete\_Rear(&de);

else

{

printf("Enter element to be inputted: ");

scanf("%d", &ele);

Insert\_Rear(&de, ele);

}

break;

default:

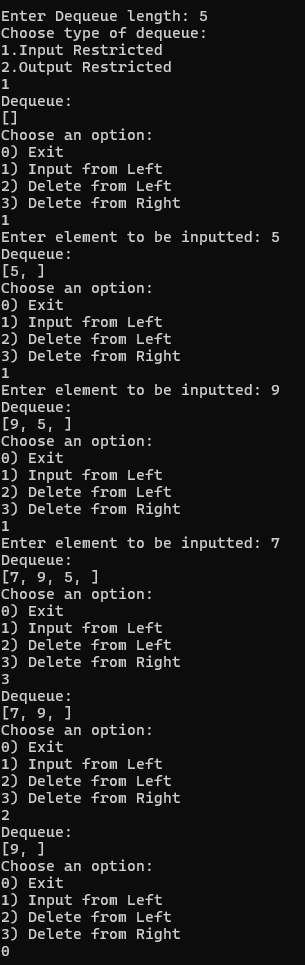
return 0;

}

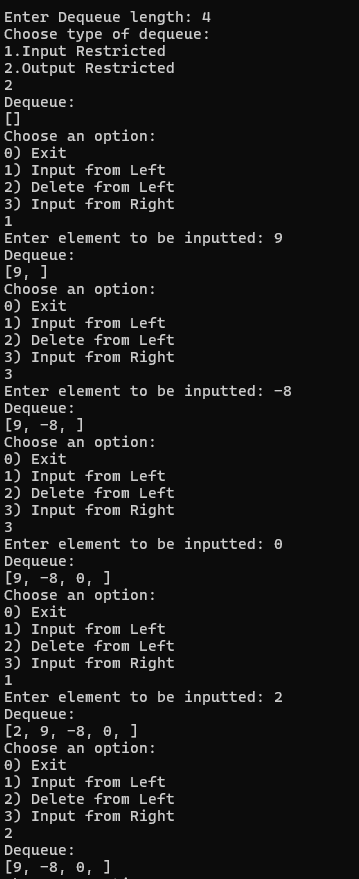
}

}

**Output: i) Input Restricted**

****

**ii) Output Restricted**

****

**Submitter Details:-**

**Name:** Faizan Dodiya

**Roll No:** 24

**Div/Batch :** A/ S-2