DOUBLE ENDED QUEUE (DEQUEUE)

AIM:

To write a C program to implement a double ended queue (dequeue) with all possible operations on it.

ALGORITHM:

1. Initialize all variables and functions.
2. Read choices.
3. If queue is not full, add items at front or back. And increment top value by 1.
4. Else print "Queue is full".
5. If Queue is not empty, delete items at front or back. And decrement top value by 1.
6. Else print "Queue is empty".
7. Display the queue items.
8. End of the program.

```
**** Program for Double Ended Queue ****


#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#define size      5

struct

dequeue{ int

deq[size];

int

front,rear;

}Q; int

Qfull() {

if(Q.rear==s
```

```
ize-1)
return 1;
else return
0;
}

int Qempty() { if((Q.front>Q.rear) ||
(Q.front==-1 && Q.rear==-1)) return 1; else
return 0; } int insert_rear(int item)
{ if(Q.front==-1 &&
Q.rear==-1)
Q.front++;
Q.deq[++Q.rear]
=item; return
Q.rear; } int
del_front() {
int item;
if(Q.front==-1)
Q.front++;
item=Q.deq[Q.fr
ont];
Q.deq[Q.front]=-1;
Q.front++; return
item; } int
insert_front(int
item)
{ int
i,j;
```

```
if(Q.fron
t==-1)
Q.front++
;
i=Q.front
-1;
while(i>=
0)
{
Q.deq[i+1]=Q.d
eq[i]; i--; }
j=Q.rear;
while(j>=Q.fro
nt) {
Q.deq[j+1]=Q.d
eq[j]; j--; }
Q.rear++;
Q.deq[Q.front]=item;
return Q.front; } int
del_rear() { int item;
item=Q.deq[Q.rear];
Q.deq[Q.rear]=-1;
Q.rear--; return item;
} void display() { int
i; printf("\n Queue is
");
for(i=Q.front;i<=Q.rea
r;i++) printf(" %d
```

```
",Q.deq[i]); } void

main()

{int choice,i,item;
Q.front=-1; Q.rear=-1; for(i=0;i<size;i++)

Q.deq[i]=-1; clrscr(); printf("\n\n\n\n

Double ended queue OR Dequeue\n"); do

{printf("\n1.Insert   front\n2.Insert   rear\n3.Delete   front\n4.Delete

rear\n"); printf("5.Display\n6.Exit\n"); printf("\nEnter ur choice:");

scanf("%d",&choice);      switch(choice)      {case      1:if(Qfull())

printf("\nDequeue is  full"); else

{printf("\nEnter item to be inserted:");

scanf("%d",&item);

Q.front=insert_front(it

em);    }break;    case

2:if(Qfull())

printf("\nDequeue    is

full"); else

{printf("\nEnter item to be inserted:");

scanf("%d",&item);

Q.rear=insert_rear(item)

;      }break;      case

3:if(Qempty())

printf("\nDequeue     is

empty");      else      {

item=del_front();

printf("\nThe item deleted from queue is  %d",item);

}break;  case  4:if(Qempty())  printf("\nDequeue

is      empty");   else   {   item=del_rear();
```

```
printf("\nThe item deleted from queue is
%d",item);
} break;
case
5:display()
; break;
case
6:exit(0);
}
}while(choice!=6);
getch();
}
```

OUTPUT:

1.Insert front

2.Insert rear

3.Delete front

4.Delete rear

5.Display

6.Exit

Enter ur choice:1

Enter item to be inserted:11

1.Insert front

2.Insert rear

3.Delete front

4.Delete rear

5.Display6.Exit

Enter ur choice:2

The item deleted from queue is 12

1.Insert front

2.Insert rear

3.Delete front

4.Delete rear

5.Display

6.Exit

Enter ur choice:5

Queue is 11 12

1.Insert front

2.Insert rear

3.Delete front

4.Delete rear

5.Display

6.Exit

Enter ur choice:1

Enter item to be inserted:10

1.Insert front

2.Insert rear

3.Delete front

4.Delete rear

5.Display

6.Exit

Enter ur choice:2

Enter item to be inserted:13

1.Insert front

2.Insert rear

3.Delete front

4.Delete rear

5.Display

6.Exit

Enter ur choice:5

Queue is 10 11 12 13

1.Insert front

2.Insert rear

3.Delete front

4.Delete rear

5.Display

6.Exit

Enter ur choice:3

The item deleted from queue is 10

1.Insert front

2.Insert rear

3.Delete front

4.Delete rear

5.Display

6.Exit

Enter ur choice:4

The item deleted from queue is 13

1.Insert front

2.Insert rear

3.Delete front

4.Delete rear

5.Display

6.Exit

Enter ur choice:5

Queue is 11 12

1.Insert front

2.Insert rear

3.Delete front

4.Delete rear

5.Display

6.Exit

Enter ur choice:6

RESULT:

Thus the C program for Double ended queue is implemented and insertion, deletion on both ends is done successfully.