

**EX. NO: 1B****QUEUE ADT USING ARRAY****AIM:**

To write a program for Queue using array implementation.

**DESCRIPTION:**

A queue data structure can be implemented using one dimensional array. But, queue implemented using array can store only fixed number of data values. The implementation of queue data structure using array is very simple, just define a one dimensional array of specific size and insert or delete the values into that array by using FIFO (First In First Out) principle with the help of variables 'front' and 'rear'. Initially both 'front' and 'rear' are set to -1. Whenever, we want to insert a new value into the queue, increment 'rear' value by one and then insert at that position. Whenever we want to delete a value from the queue, then increment 'front' value by one and then display the value at 'front' position as deleted element.

**ALGORITHM:**

1. Define a array which stores queue elements..
2. The operations on the queue are
  - a. a)INSERT data into the queue
  - b. b)DELETE data out of queue
3. INSERT DATA INTO queue
  - a. Enter the data to be inserted into queue.
  - b. If TOP is NULL
    - i. The input data is the first node in queue.
    - ii. The link of the node is NULL.
    - iii. TOP points to that node.
  - c. If TOP is NOT NULL
    - i. The link of TOP points to the new node.
    - ii. TOP points to that node.
4. DELETE DATA FROM queue
  - a. If TOP is NULL
    - i. the queue is empty
  - b. If TOP is NOT NULL

- i. The link of TOP is the current TOP.
  - ii. The pervious TOP is popped from queue.
5. The queue represented by linked list is traversed to display its content.

### **PROGRAM**

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h> #define
SIZE 5 int front = - 1; int
rear = - 1; int q[SIZE];
void insert( ); void del( );
void display( ); void main(
) { int choice; do {
printf("\t Menu");
printf("\n 1. Insert");
printf("\n 2. Delete");
printf("\n 3. Display ");
printf("\n 4. Exit");
printf("\n Enter Your
Choice:"); scanf("%d",
&choice); switch(choice) {
case 1:
insert( ); display( ); break;
case 2:
del( ); display( ); break;
case 3:display( );
```

```

    break; case 4: printf("End
of Program....!!!!");
exit(0); }}while(choice !=
4);} void insert( ) { int no;
printf("\n Enter No.:" );
scanf("%d", &no); if(rear <
SIZE - 1) { q[++rear]=no;
if(front == -1) front=0;//
front=front+1;
} else { printf("\n
Queue overflow");
}} void del( ) {
if(front == - 1) {
printf("\n Queue
Underflow"); return; }
else

{ printf("\n Deleted Item:-->%d\n",
q[front]);
} if(front ==
rear)
{ front
= - 1;
rear = -
1; }
else
{front = front + 1;

```

```

}} void

display( ) {

int i; if(

front == - 1)

{ printf("\nQueue is

empty...."); return; }

for(i = front; i<=rear;

i++) printf("\t%d",q[i]);}

```

## OUTPUT

```

E:\DESKTOP\DS LAB CS8381\queue\bin\Debug\queue.exe
Menu
1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice:1

Enter No.:23
23 Menu
1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice:45
45 Menu
1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice:1

Enter No.:56
23 56 Menu
1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice:

```

```
"E:\DESKTOP\DS LAB CS8381\queue\bin\Debug\queue.exe"
2. Delete
3. Display
4. Exit
Enter Your Choice:3
    23    56    Menu
1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice:2

Deleted Item:-->23
    56    Menu
1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice:4
End of Program.....!!!!
Process returned 0 (0x0)   execution time : 79.546 s
Press any key to continue.
```

## **RESULT:**

Thus a C program for Queue using ADT was implemented successfully