PROGRAM 17

AIM: - TO IMPLEMENT CIRCULAR QUEUE

ALGORITHM

INSERT (FRONT, REAR, MAX, ITEM)

Step 1:-IF FRONT = REAR – 1 | |FRONT = 0 && REAR = N – 1

Print Overflow return

Step 2:-Read data to insert

Step 3:-IF FRONT = -1

Set FRONT = 0

Set REAR = 0

Go to step 5

Step 4:- IF REAR = N – 1 && FRONT! = 0 then

REAR = 0

ELSE

REAR = REAR +1

Step 5:-QUEUE [REAR] = DATA

Step 6:-End

DELETE (FRONT, REAR, MAX, ITEM)

Step 1:-IF FRONT = =-1 then

Print Underflow return

Step 2:-Set ITEM = = QUEUE [FRONT]

Step 3:-IF FRONT = = REAR

Set FRONT =-1

Set REAR =-1

Goto step 4

Step 4:-IF FRONT = N-1 then

Set FRONT =0

ELSE

SET FRONT = FRONT +1

Step 5:-End

SOURCE CODE

#include<stdio.h>

#include<stdlib.h>

#define MAX 3

int FRONT =-1;

int REAR =-1;

int QUEUE[MAX];

int in(int []);

int del(int []);

int display(int []);

void main()

{

int a;

system("COLOR F0");

printf("1.Insert");

printf("\n2.Delete");

printf("\n3.Display");

printf("\n4.Exit");

do

{

printf("\nEnter the Choice");

scanf("%d",&a);

switch(a)

{

case 1:in(QUEUE);

break;

case 2:del(QUEUE);

break;

case 3:display(QUEUE);

break;

case 4:exit(0);

break;

default:printf("\nWrong Choice");

}

}while(1);

}

int in(int QUEUE[MAX])

{

int item;

if(FRONT==0&&REAR==MAX-1)

{

printf("\nOverflow");

return;

}

if(REAR==MAX-1&&FRONT!=0)

{

REAR=0;

}

else

{

REAR=REAR+1;

}

if(FRONT==-1)

{

FRONT=0;

REAR=0;

}

printf("\nEnter the item");

scanf("%d",&item);

QUEUE[REAR]=item;

return 0;

}

int del(int QUEUE[MAX])

{

int item;

if(FRONT==-1)

{

printf("\nunderflow");

return;

}

item=QUEUE[FRONT];

if(FRONT==MAX-1)

{

FRONT=0;

}

else

{

FRONT=FRONT+1;

}

if(FRONT==REAR)

{

FRONT=-1;

REAR=-1;

}

}

int display(int QUEUE[MAX])

{

int i;

if(FRONT<REAR)

{

for(i=FRONT;i<=REAR;i++)

{

printf("%d",QUEUE[i]);

}

}

else

{

for(i=FRONT;i<MAX;i++)

{

printf("%d",QUEUE[i]);

}

for(i=0;i<=REAR;i++)

{

printf("%d",QUEUE[i]);

} } }



