Program 16

Aim- To implement Queue

Algorithm

INSERT (QUEUE, MAX, FRONT, REAR, ITEM)

Step 1:-[Check the overflow condition]

IF REAR= =MAX - 1 then

Print Overflow and end.

Step 2:-[Check 0 element condition]

IF FRONT = =-1 then

Set FRONT = 0 Set REAR = 0

ELSE

REAR = REAR +1

Step 3:-[Element insert at rear position]

Queue [REAR] = ITEM

Step 4:-End

DELETE (QUEUE, MAX, FRONT, REAR, ITEM)

Step 1:-IF FRONT = = -1 then

Print Underflow and exit

Step 2:-Set ITEM = QUEUE [FRONT]

Step 3:-IF FRONT= =REAR

Then set FRONT = = -1

Then set REAR = = -1

ELSE

FRONT = FRONT +1

Step 4:-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(REAR==MAX-1)

{

printf("\nOverflow");

return;

}

if(FRONT==-1)

{

FRONT=0;

REAR=0;

}

else

{

REAR=REAR+1;

}

printf("\nEnter the item");

scanf("%d",&item);

QUEUE[REAR]=item;

return 0;

}

int del(int QUEUE[MAX])

{

int item;

if(FRONT==-1)

{

printf("\nunderflow");

}

item=QUEUE[FRONT];

if(FRONT==REAR)

{

FRONT=-1;

REAR=-1;

}

else

{

FRONT=FRONT+1;

} }

int display(int QUEUE[MAX])

{ int i;

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

{ printf("%d\n",QUEUE[i]); } }



